МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ
Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Пензенский государственный университет» (ПГУ)
Российский фонд фундаментальных исследований

НОВЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И СИСТЕМЫ

СБОРНИК НАУЧНЫХ СТАТЕЙ
XIV Международной научно-технической
конференции,
посвященной 70-летию кафедры «Вычислительная техника»
и 30-летию кафедры «Системы автоматизированного проектирования»

г. Пенза, 22–24 ноября 2017 г.

Proceedings of the Fourteenth International Conference
of Science and Technology

NEW INFORMATION TECHNOLOGIES AND SYSTEMS

dedicated to the 70th anniversary of the Department
of Computer Engineering ,
and the 30th anniversary of Computer Aided Design Department

Penza, Russia, November 22–24, 2017

Пенза
Издательство ПГУ
2017

УДК 004.4
Н72

Новые информационные технологии и системы : сб. науч. ст. Н72 XIV Междунар. науч.-техн. конф., посвящ. 70-летию кафедры «Вычислительная техника» и 30-летию кафедры «Системы автоматизированного проектирования» (г. Пенза, 22–24 ноября 2017 г.). – Пенза : Изд-во ПГУ, 2017. – 420 с.

ISBN 978-5-906975-95-9

Сборник статей подготовлен на основе докладов, сделанных на XIV Международной научно-технической конференции «Новые информационные технологии и системы» (НИТиС–2017), посвященной 70-летию кафедры «Вычислительная техника» и 30-летию кафедры «Системы автоматизированного проектирования». Конференция проведена Министерством образования и науки РФ и Пензенским государственным университетом при поддержке Российского фонда фундаментальных исследований, проект 17-07-20569.

Доклады охватывают широкий спектр проблем в области проектирования и администрирования информационных систем и систем автоматизированного проектирования. Рассматриваются вопросы управления в социальных, экономических и технических системах. Представлены современные технологии хранения и обработки данных, информационные технологии в образовании. Обсуждаются вопросы математического моделирования в научных исследованиях, а также вопросы разработки программного и аппаратного обеспечения информационно-вычислительных систем и сетей и их информационной безопасности.

Издание предназначено для инженеров и научных работников, а также обучающихся старших курсов, интересующихся вопросами современных информационных технологий.

УДК 004.4

Редакционная коллегия:
В. И. Волчихин, Д. В. Артамонов, И. И. Артемов, Д. В. Пащенко, А. М. Бершадский, И. П. Бурукина, Н. Н. Коннов

ISBN 978-5-906975-95-9

© Пензенский государственный университет, 2017

АППАРАТНО-ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ИНФОРМАЦИОННО-ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ И СИСТЕМ УПРАВЛЕНИЯ

Д. Н. Дроздов[1,2], С. Патил[2], В. Н. Дубинин[1]

[1] Пензенский государственный университет, Пенза, Россия
[2] Технический университет Люлео, Люлео, Швеция

ФОРМАЛЬНАЯ ВЕРИФИКАЦИЯ КИБЕРФИЗИЧЕСКИХ СИСТЕМ УПРАВЛЕНИЯ СО СЛУЧАЙНЫМИ ВРЕМЕННЫМИ ЗАДЕРЖКАМИ

Рассматривается способ моделирования и формальной верификации распределенных киберфизических систем управления промышленной автоматикой. Для моделирования используется язык SMV с применением определенного класса абстракций с целью сокращения пространства состояний модели.

В настоящее время в области промышленной автоматики происходит сдвиг парадигмы с централизованных систем управления к распределенным. Такие системы управления содержат множество «умных» компонентов, таких как датчики, приводы и миниатюрные контроллеры, соединенных между собой посредством проводных и беспроводных сетей передачи данных, и образующих цельную систему совместно с управляемым оборудованием. В подобных системах важной характеристикой программного обеспечения является устойчивость к различным возмущениям физической среды, одним из наиболее часто наблюдаемых результатов которых является возникновение случайных задержек пакетов в беспроводных сетях передачи данных.

Данная проблема изначально рассматривалась в рамках теории автоматического управления [1, 2], однако, в данных работах не принимается во внимание конкретная реализация управляющего программного обеспечения (ПО). В качестве идеи, призванной охватить одновременно и процессы, физической системы и особенности программной реализации, в работе [3] предлагается концепция киберфизическог агностицизма (Cyber-Physical Agnosticism, CPA). Данная концепция рассматривается как свойство ПО проявлять всегда предсказуемое поведение, не зависимое от различных возмущений в рассматриваемой киберфизической системе.

Как было продемонстрированно в работах [4–6], распределенные киберфизические системы с сервисно-ориентированной архитектурой могут эффективно моделироваться с использованием функциональных блоков стандарта IEC 61499, а в работе [7] демонстрируется использование веб-сервисов в приложениях на базе IEC 61499.

В то время, как киберфизическая система может быть промоделирована в системе Ptolemy II с применением гетерогенной комбинации дискретного и непрерывного доменов (с использованием семантики pTides), реализация подобных систем требует соответствующей модификации языка, описываемого стандартом IEC 61499, за счет расширения понятия «событие» добавлением временных меток.

В данной статье рассматривается упрощенная реализация временных меток для IEC 61499. Каждому событийному входу или выходу функционального блока, ставится в соответствие дополнительный информационный вход/выход типа TIME, предназначенный для передачи временной метки соответствующего события между функциональными блоками [8]. Использование подобной методики позволяет выполнять моделирование и даже тестирование в реальных условиях без необходимости внесения изменений в среду исполнения и разработки ПО.

Для управления событиями с переменными метками, в модель добавлен инкрементный глобальный таймер, а чтобы ограничить пространство состояний при верификации модели, рассматривается только интервал времени от 0 до $T_{\max}$. Изменение глобальной переменной времени происходит в соответствии со следующим правилом:

$$T_{Global} < T_{max} \vee \gamma \vee (\wedge(Do_i)) \Rightarrow T_{Global} = T_{Global} + D_{min}.$$

Кроме того, правила изменения локального времени блоков E_DELAY/E_CYCLE, приведенные в [9], были модифицированы таким образом, чтобы останавливать выполение при достижении глобального времени $T_{\max}$.

Предложенный подход протестирован на примере цилиндра с PID-регулятором, приведенном в работе [3]. На рис. 1 приведено схематичное изображение данного примера. Контроллер пневматического цилиндра с возвратной пружиной реализован в виде управляющего приложения на базе функциональных блоков.

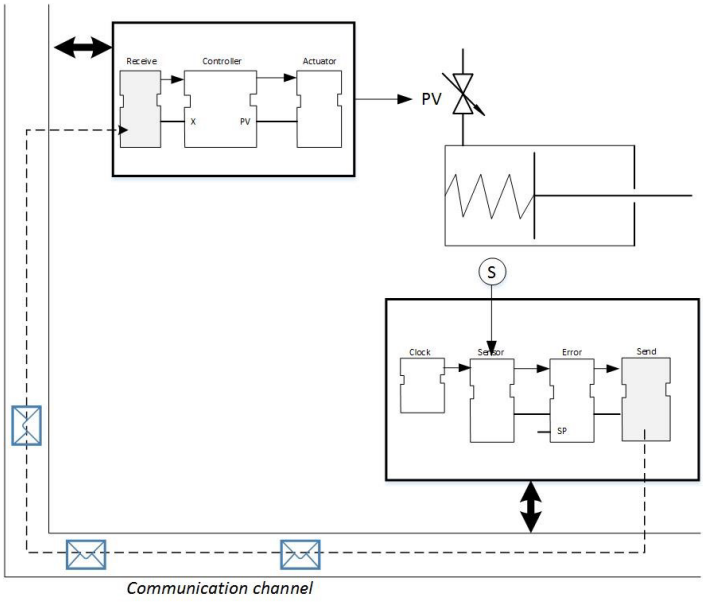
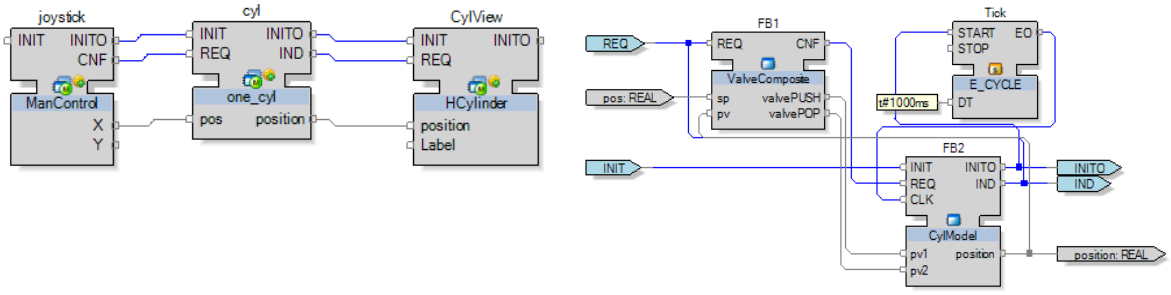


Рис. 1. Демонстрационный пример: цилиндр с PID-регулятором

На рис. 2,*а* показана модель верхнего уровня в системе nxtStudio, состоящая из джойстика, модели цилиндра с контроллером, и визуализации цилиндра, активного в режиме имитационного моделирования. На рис. 2,*б* изображена замкнутая модель цилиндра с контроллером.

Для формальной верификации данной системы использовался верификатор nuSMV. Основа SMV-модели контроллера сгенерирована с помощью утилиты fb2smv из управляющего ПО и модифицирована и дополнена как показано ниже.



(*а*) (*б*)

Рис. 2. Система верхнего уровня в среде nxtStudio (*а*) и замкнутая модель цилиндра с контроллером (*б*)

В первую очередь, в имитационной модели в среде nxtStudio, были созданы два CAT-блока: GetTimestamp и TimeDiff, отвечающие за генерацию временной метки события и вычисления разницы между временной меткой и текущим временем системы, а так же добавлен блок E_DELAY, эмулирующий задержку передачи данных в сети. Эмуляция случайной задержки в беспроводных сетях передачи данных обеспечивается за счет подачи на вход блока E_DELAY случайной величины задержки при подаче события на вход.

Полученная после генерации SMV-кода система дополняется SMV-моделью оборудования и модифицированными временным диспетчерами. Для сокращения пространства состояний модели, непрерывное изменение положения поршня в управляемом цилиндре дискретизируется по времени с шагом в 500 единиц модельного времени. График изменения положения поршня, полученный из контрпримера, сгенерированного системой nuSMV, приведен на рис. 3.

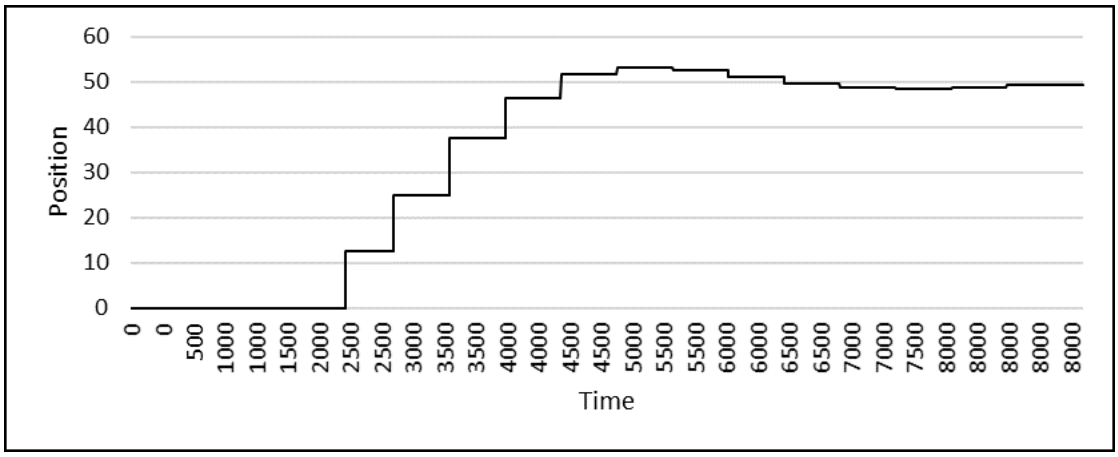


Рис. 3. График положения поршня в цилиндре, извлеченный из контрпримера nuSMV

Библиографический список

1. Nilsson, J. Analysis of real-time control systems with time delays / J. Nilsson, B. Bernhardsson // Decision and Control : Proceedings of the 35th IEEE Conference on. – IEEE, 1996. – Т. 3. – С. 3173–3178.

2. Nilsson, J. Real-time control systems with delays / J. Nilsson et al. ; Lund Institute of Technology. – Lund, 1998. – 138 p.

3. Vyatkin, V. Towards cyber-physical agnosticism by enhancing IEC 61499 with ptides model of computations / V. Vyatkin, C. Pang, S. Tripakis // Industrial Electronics Society : IECON 2015-41st Annual Conference of the IEEE. – IEEE, 2015. – С. 001970–001975.

4. Dai, W. Bridging service-oriented architecture and IEC 61499 for flexibility and interoperability / W. Dai et al. // IEEE Transactions on Industrial Informatics. – 2015. – Т. 11, № 3. – С. 771–781.

5. Dai, W. Service-oriented distributed control software design for process automation systems / W. Dai et al. // Systems, Man and Cybernetics (SMC) : 2014 IEEE International Conference on. – IEEE, 2014. – С. 3637–3642.

6. Dai, W. Function block implementation of service oriented architecture: Case study / W. Dai et al. // Industrial Informatics (INDIN) : 2014 12th IEEE International Conference on. – IEEE, 2014. – С. 112–117.

7. Demin, E. Automation services orchestration with function blocks: Web-service implementation and performance evaluation / E. Demin et al. // Service Orientation in Holonic and Multi-Agent Manufacturing. – Springer International Publishing, 2016. – С. 213–221.

8. Drozdov, D. Formal Modelling of Distributed Automation CPS with CP-Agnostic Software / D. Drozdov, S. Patil, V. Vyatkin // Service Orientation in Holonic and Multi-Agent Manufacturing. – Springer, Cham, 2017. – С. 35–46.

Д. Н. Дроздов[1,2], С. Патил[2]

[1] Пензенский государственный университет, Пенза, Россия
[2] Технический университет Люлео, Люлео, Швеция

ПОСТРОЕНИЕ РАСПРЕДЕЛЕННЫХ СИСТЕМ УПРАВЛЕНИЯ ПРОМЫШЛЕННОЙ АВТОМАТИКОЙ НА ОСНОВЕ СТАНДАРТА IEC 61499: ДЕКОМПОЗИЦИЯ И ОТЛАДКА

Активно развиваемый в настоящее время стандарт для построения систем управления промышленной автоматикой IEC 61499 предлагает новый подход к построению распределенных динамически реконфигурируемых систем управления промышленной автоматикой. Данная статья на реальном примере показывает возможности декомпозиции сложного управляющего автомата на множество простых функциональных блоков в рамках стандарта IEC 61499, а также касается вопросов отладки подобных распределенных систем.

Активно развиваемый в настоящее время стандарт для построения систем управления промышленной автоматикой IEC 61499 предлагает новый подход к построению распределенных динамически реконфигурируемых систем управления промышленной автоматикой [1], а с применением последних решений, – интеграцию идеологии интернета вещей для построения гибких производственных киберфизических систем. Кроме того, ряд работ направлен на создание и адаптацию различных методов верификации подобных систем [2–4].

Однако, старый подход к разработке централизованного управляющего ПО за десятки лет доминирования стандарта IEC 61131-3 прочно укоренился в головах разработчиков. Данная статья на реальном примере показывает возможности декомпозиции сложного управляющего автомата на множество простых функциональных блоков в рамках стандарта IEC 61499, а также преимущество данного подхода для отладки и построения реконфигурируемых систем.

В качестве исходного примера для демонстрации подхода был взят модуль DistributionStation учебной производственной системы FESTO MPS 500. Внешний вид станции приведен на рис. 1. Станция состоит из магазина заготовок с выталкивателем и руки-маятника с присоской, выполняющей захват заготовки и передачу ее к следующей станции, а также ряда двигателей и датчиков.

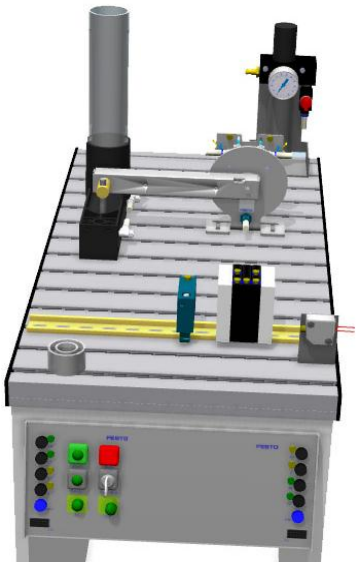


Рис. 1. Внешний вид (3D-модель) модуля DistributionStation

Управляющее ПО данной станции представлено как компонент общей системы управления сборочной линии FESTO MPS 500. Данный компонент, помимо прочего, осуществляет коммуникацию с тестирующей станцией (которой передается заготовка) по линиям передачи данных. На рис. 2 изображена диаграмма функциональных блоков верхнего уровня управляющего ПО всей системы.

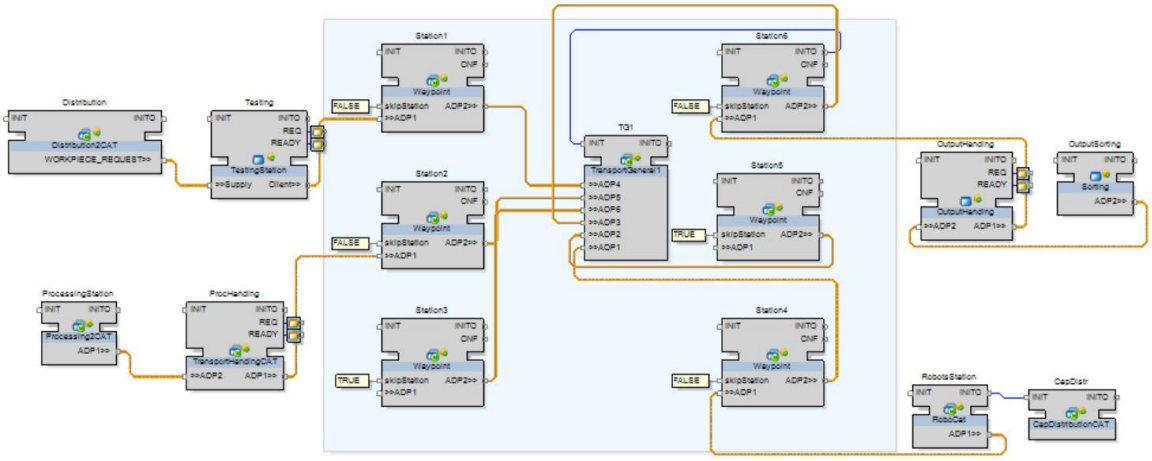


Рис. 2. Диаграмма функциональных блоков управляющего ПО системы MPS 500

Классический подход к созданию системы управления отдельным модулем основывается на создании одного управляющего автомата с множеством состояний. Такой автомат продемонстрирован на рис. 3. Данный подход позволяет быстро создать прототип управляющего ПО, однако, дальнейшая отладка и модификация оказывается затруднительной.

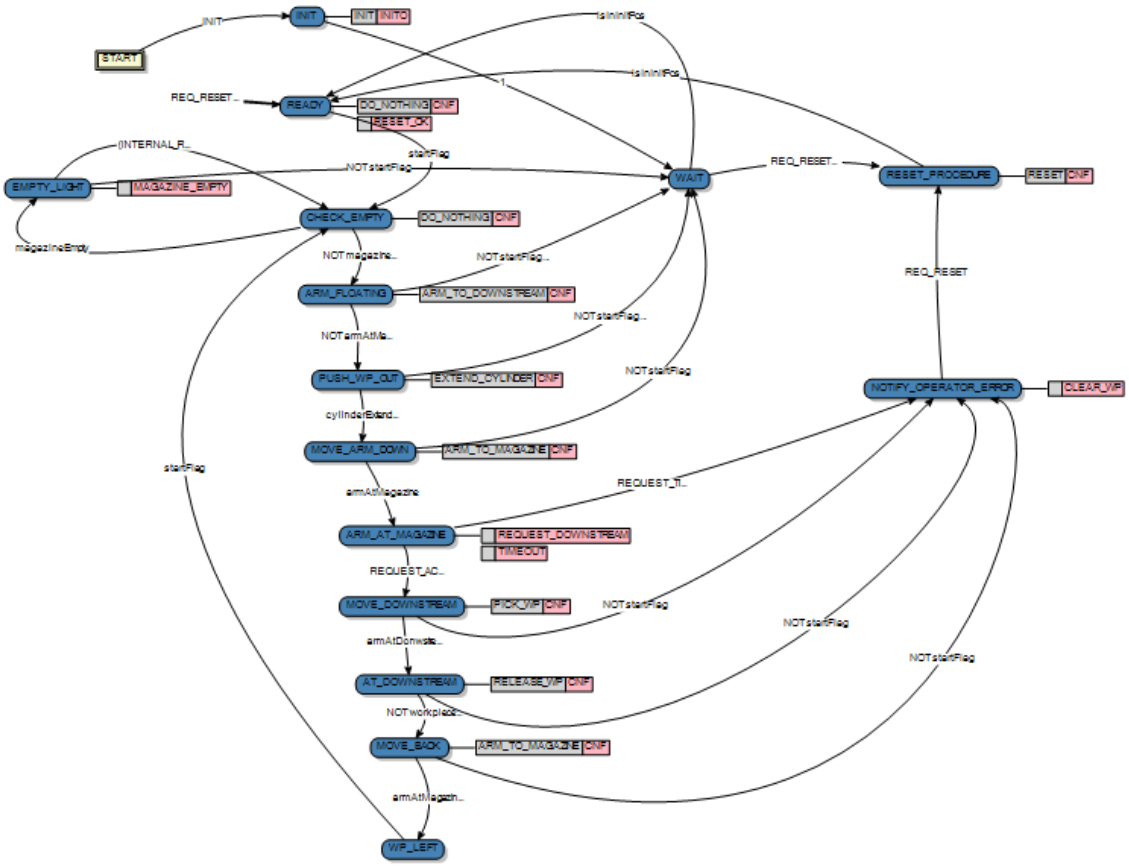


Рис. 3. Управляющий автомат (в форме диаграммы ECC) модуля DistributionStation

Так как все реализации стандарта IEC 61499 предоставляют развитые средства для отладки распределенной блочной событийно-управляемой архитектуры, целесообразной оказывается декомпозиция управляющего блока на множество отдельных, событийно-управляемых блоков, выполняющих простейшие операции.

Первое, что необходимо выделить при такой декомпозиции – модули преобразования управляющих событийных сигналов в логические уровни на портах ввода-вывода контроллера (условно назовем их драйверами механических устройств). В рассматриваемом примере было выделено три таких драйвера, названных: CustomStopActuator, SingleActingCylinder и ArmReset. Фактически, оба драйвера ArmReset и CustomStopActuator управляют одним и тем же элементом оборудования – рукой-маятником, однако, драйвер ArmReset выполняем специфическую процедуру установки первоначальной позиции, поэтому его было решено реализовать как отдельный блок. На рис. 4 изображены вышеописанные блоки драйверов с соответствующими управляющими автоматами.

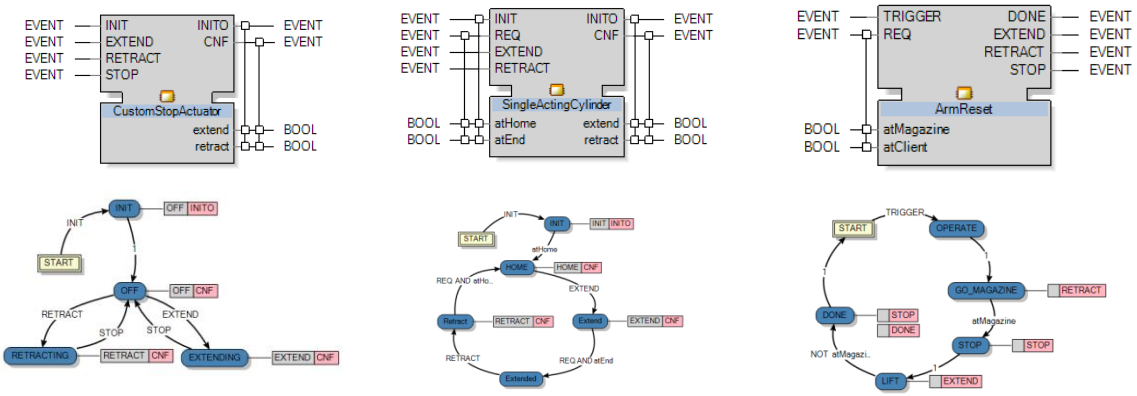


Рис. 4. Блоки «драйверов» механических устройств

Основная управляющая программа была разбита на серию простейших действий, каждое из которых выполняется специальным блоком-агентом, активизирующемся при поступлении сигнала на событийный вход TRIGGER и выдающим событийный сигнал DONE после окончания исполнения. Главная управляющая последовательность модуля DistributionStation изображена на рис. 5 (соединительные линии входных и выходных сигналов от оборудования/драйверов были удалены, чтобы не загромождать рисунок). Как можно увидеть, управляющая последовательность состоит из шести действий: GetWP (получить заготовку из магазина), ToMagazine (переместить руку-маятник в сторону магазина), Grab (захватить заготовку используя управляемую присоску), ToClient (переместить руку-маятник в пул для отгрузки заготовку), Release (используя присоску, отпустить заготовку в пул отгрузки), Reset (вернуть руку-маятник в первоначальное положение).

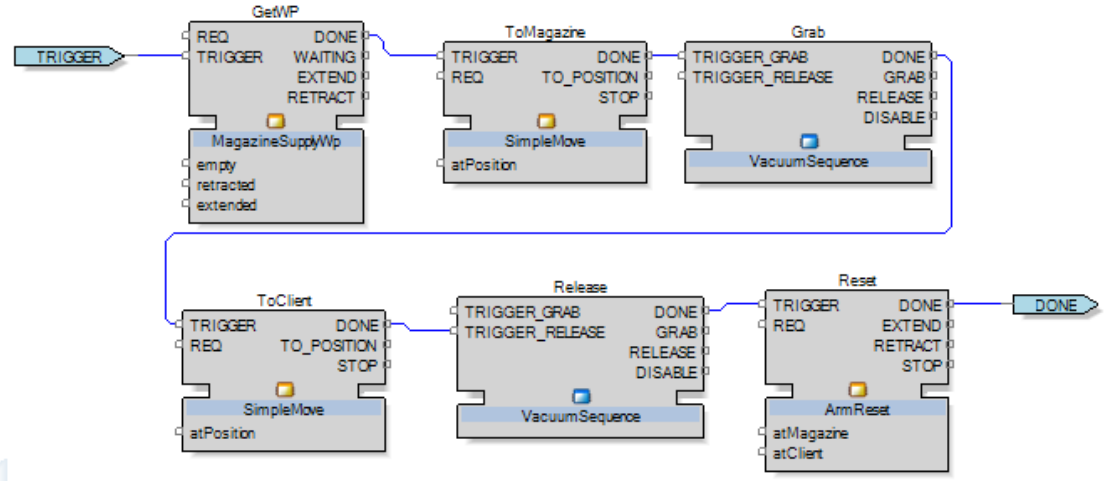


Рис. 5. Главная управляющая последовательность модуля DistributionStation

Каждый из блоков в данной последовательности имеет в своей основе простой управляющий автомат наподобие изображенных на рис. 4. Подобный подход, за счет контроля входных и выходных событий блоков, позволяет с легкостью выполнять отладку сначала отдельных простейших действий, а позже и комбинированных цепочек. Каждая цепочка действий, наподобие приведенной на рис. 5, заключается внутрь составного функционального блока и такой блок уже сам может выступать как реализация отдельного действия в более сложной комбинированной программе.

Библиографический список

1. Vyatkin, V. IEC 61499 as Enabler of Distributed and Intelligent Automation: State-of-the-Art Review / V. Vyatkin // IEEE transactions on industrial informatics. – 2011. – Т. 7, № 4. – С. 768–781.
2. Дубинин, В. Н. Модели функциональных блоков IEC 61499, их проверка и трансформации в проектировании распределенных систем управления : моногр. / В. Н. Дубинин, В. В. Вяткин. – Пенза : Изд-во ПГУ, 2012. – 348 с.
3. Patil, S. Cloud-Based Framework for Practical Model-Checking of Industrial Automation Applications / S. Patil, D. Drozdov, V. Dubinin, V. Vyatkin // DoCEIS. – 2015. – С. 73–81.
4. Drozdov, D. Formal verification of cyber-physical automation systems modelled with timed block diagrams / D. Drozdov et al. // Industrial Electronics (ISIE) : IEEE 25th International Symposium on. – IEEE, 2016. – С. 316–321.

Д. Н. Дроздов[1,2], С. Патил[2]

[1] Пензенский государственный университет, Пенза, Россия
[2] Технический университет Люлео, Люлео, Швеция

РАСПРЕДЕЛЕННАЯ СИСТЕМА УПРАВЛЕНИЯ ПРОИЗВОДСТВЕННОЙ ЛИНИЕЙ FESTO MPS-500

В рамках развития систем управления промышленной автоматикой важную позицию занимает переход от классических централизованных систем к распределенным. Серьезный толчок в данном направлении продиктован концепцией Промышленности 4.0. В данной статье описывается пример распределенной системы управления, разработанной для учебной производственной линии FESTO MPS-500 на базе лаборатории AIC[3] Технического Университета Люлео (Швеция).

Современные тенденции развития в области промышленной автоматики, продиктованные концепцией Промышленности 4.0 [1] требуют создания совершенно новых интеллектуальных систем управления [2]. Стандарт IEC 61499 для создания распределенного программного обеспечения (ПО) систем промышленной автоматики предлагает частичное решение этой проблемы на программном уровне. Однако, новая парадигма разработки ПО накладывает определенные требования к аппаратной платформе. Системы управления, созданные на базе предыдущего стандарта IEC 61131-3 были в высокой степени централизованным и жестко программировались под один сценарий поведения, при этом возможность быстрой реконфигурации системы никогда не рассматривалась и в аппаратной части было достаточно ПЛК, жестко «привязанного» к оборудованию и инфраструктуре. Новые системы управления на базе IEC 61499 предлагают беспрецедентные возможности [3], позволяющие, после изменения конфигурации производственного оборудования, выполнять реконфигурацию уже на программном уровне всего в пару щелчков мыши. Это становится возможным благодаря использованию множества «умных» аппаратных компонентов со встроенным контроллером, заранее запрограммированным для выполнения базовых операций.

Реконфигурируемые системы промышленной автоматики, которые должны работать в рамках новых «умных» фабрик, подразумевают возможность быстрой перекомпоновки системы. На программном уровне, как уже было сказано выше, данный вопрос решается созданием управляющего ПО с использованием распределенных стандартов программирования, например, IEC 61499, и внедрением динамических сценариев реконфигурации. Аппаратная составляющая управляющей системы, в то же время, требует декомпозиции на максимально автономные управляющие блоки, способные перемещаться вместе с управляемым оборудованием при физической реконфигурации системы.

Структура разработанного прототипа такого управляющего блока приведена на рис. 1. Он состоит из программируемого логического контроллера (ПЛК) со встроенными блоками цифрового ввода-вывода (БВВ), блока питания (БП), WiFi адаптера и дополнительного блока цифрового ввода-вывода, соединенного с ПЛК посредством интерфейса Ethercat.

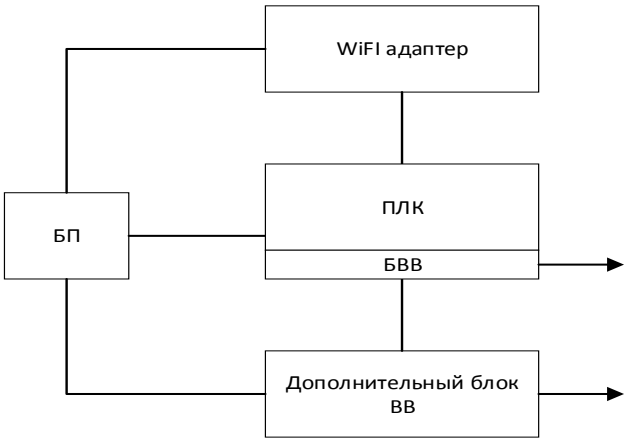


Рис. 1. Общая структура управляющего блока

В качестве программируемого логического контроллера использован миниатюрный ПЛК nxtControl DCSmini, поддерживающий управляющие программы на на базе стандарта IEC 61499. Данный ПЛК имеет сетевой интерфейс Ethernet для загрузки программ и соединения с другими ПЛК и иными компонентами управляющей системы, однако для поддержки большей мобильности, в управляющий блок включен адаптер Ethernet->WiFi, обеспечивающий подключение к беспроводным сетям передачи данных. Внешний вид собранного блока приведен на рис. 2.

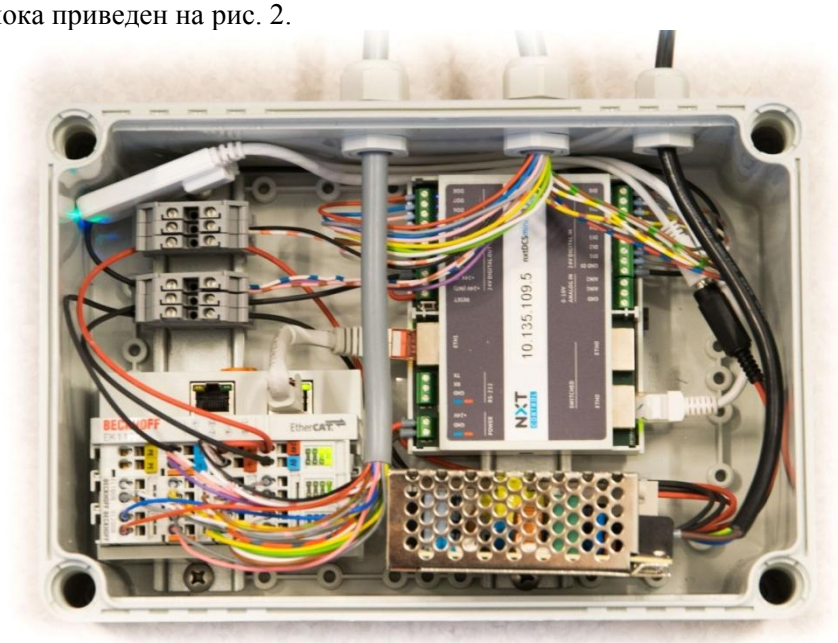


Рис. 2. Внешний вид управляющего блока

Данные блоки использованы для создания системы управления учебной производственной линией FESTO MPS-500, являющейся одной из подсистем лаборатории AIC[3][4] Технического университета Люлео (Швеция). Тестовая система, изображенная на рис. 3, состоит из конвейера, станции распределения заготовок, станции тестирования заготовок, двух манипуляторов, станции обработки деталей, станции сортировки деталей, станции взаимодействия со сборочным роботом, беспроводного маршрутизатора и рабочего местом программиста и оператора.



Рис. 3. Схема учебной производственной системы

Каждая станция была оснащена собственным управляющим блоком, а также для конвейера был создан отдельный управляющий блок без дополнительного блока ввода-вывода, но с более мощным ПЛК. В общей сложности, распределенная система управления состоит из 8 управляющих блоков, соединенных по сети WiFi между собой и с рабочим местом программиста/оператора, на которые было развернуто программное обеспечение, созданное по стандарту IEC 61499. На рис. 4 приведена диаграмма функциональных блоков верхнего уровня для управляющего ПО сборочной линии. В случае применения классической централизованной схемы управления с применением только проводных интерфейсов, при перемещении станций сборочной линии для изменения сценария работы, потребовалась бы трудоемкая работа по переподключению множества соединений и изменению управляющего ПО, однако с применением нового подходя, требуется лишь переключение питания станций и минимальные изменения в программе.

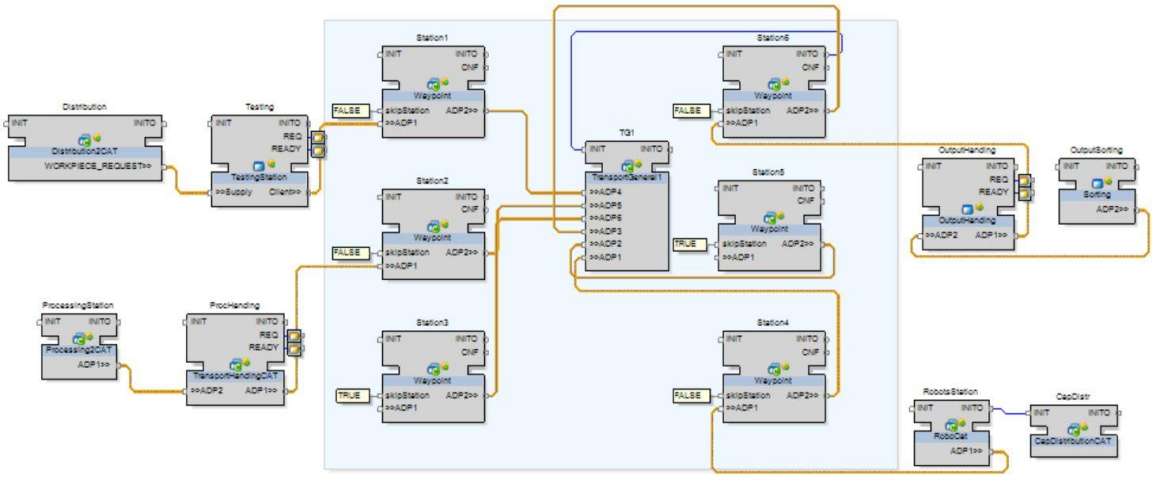


Рис. 4. Диаграмма функциональных блоков управляющего ПО сборочной линии

Заключение

В данной статье представлен прототип управляющего блока с беспроводным интерфейсом для применения в распределенных системах управления промышленной авто-

матикой с программным обеспечением на базе стандарта IEC 61499. Действенность подхода продемонстрирована на примере лабораторной сборочной линии.

Современные тенденции развития систем управления промышленной автоматикой требуют создания малогабаритных контроллеров с низким энергопотреблением. Множество наработок из области встроенной техники, однако, до сих пор не нашли применения в системах промышленной автоматики. Использование малогабаритных беспроводных контроллеров в то же время, позволит распределенным системам управления выйти на совершенно новый уровень – интернета вещей в области промышленной автоматики, где «умные» компоненты оборудования могут собираться в целые производственные линии и фабрики. Последующая работа направлена на разработку подобных малогабаритных контроллеров с беспроводными интерфейсами и поддержкой программ стандарта IEC 61499.

Библиографический список

1. Lasi, H. Industry 4.0 / H. Lasi et al. // Business & Information Systems Engineering. – 2014. – Т. 6, №. 4. – С. 239–242.

2. Monostori, L. Cyber-physical production systems: Roots, expectations and R&D challenges / L. Monostori // Procedia CIRP. – 2014. – Т. 17. – С. 9–13.

3. Vyatkin, V. IEC 61499 as Enabler of Distributed and Intelligent Automation: State-of-the-Art Review / V. Vyatkin // IEEE transactions on industrial informatics. – 2011. – Т. 7, № 4. – С. 768–781.

4. AIC^{3}lab: Automation, Industrial Computing, Communication and Control. – Luleå University of Technology, LTU. – URL: https://www.ltu.se/AICcube

Р. А. Бикташев, Н. П. Вашкевич

Пензенский государственный университет, Пенза, Россия

СТРУКТУРНАЯ РЕАЛИЗАЦИЯ И МОДЕЛИРОВАНИЕ АЛГОРИТМОВ ПЛАНИРОВАНИЯ/ДИСПЕТЧЕРИЗАЦИИ В ПАРАЛЛЕЛЬНЫХ СИСТЕМАХ НА ОСНОВЕ ЛОГИКИ НЕДЕТЕРМИНИРОВАННЫХ АВТОМАТОВ

Предложена структура аппаратного планировщика/диспетчера задач в составе многопроцессорной системы, его модели на языке недетерминированных автоматов и языке VHDL. Проведены моделирование и оценка результатов.

Известно, что системы реального времени требуют высокой реакции на запрос, а также гарантированного времени ответа. Это обстоятельство вызывает необходимость повышения производительности операционной системы, особенно систем жесткого реального времени.

Структурная реализация функций многопроцессорных операционных систем заключается в перенесении трудоемких компонент планирования и диспетчеризации задач на уровень специализированного устройства управления (процессора). Такой перенос позволяет освободить программную часть ОС от функций управления переключением задач и прерываниями. В результате сокращается время выполнения соответствующих API-функций, а тщательная отладка аппаратуры современными средствами проектирования ПЛИС гарантирует надежность и безопасность операционной системы. Аппаратный подход позволяет также уменьшить разброс времени (джиттер) выполнения процессов по сравнению с программной реализацией, что значительно повышает гарантийные сроки выполнения приложений реального времени.

Существуют два подхода к реализации планировщиков/диспетчеров задач: с глобальной очередью и с локальными очередями в каждом процессорном узле. Аппаратная

реализация первого из них достаточно подробно рассмотрена в [1], структурная реализация второго представлена на рис. 1.

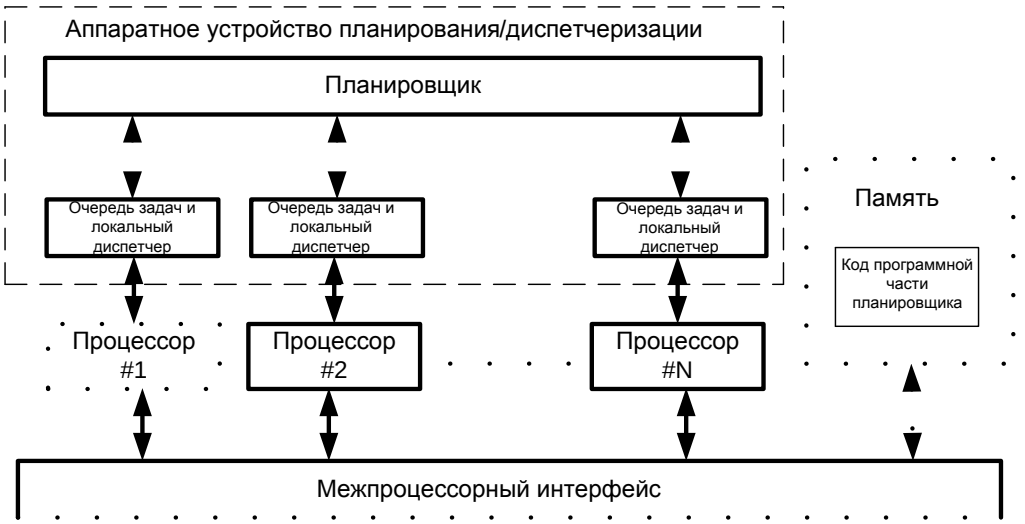


Рис. 1. Схема включения планировщика/диспетчера задач в состав многопроцессорной системы

В аппаратной части можно выделить два основных типа блока: собственно планировщик задач и локальную очередь задач с диспетчером. Первый принимает извне (например, из внешней памяти) новую задачу, помещает ее в одну из очередей, после чего в работу включается локальный диспетчер.

Интерфейс взаимодействия устройства локального диспетчера с каждым из процессоров и с планировщиком задач представлен на рис. 3.

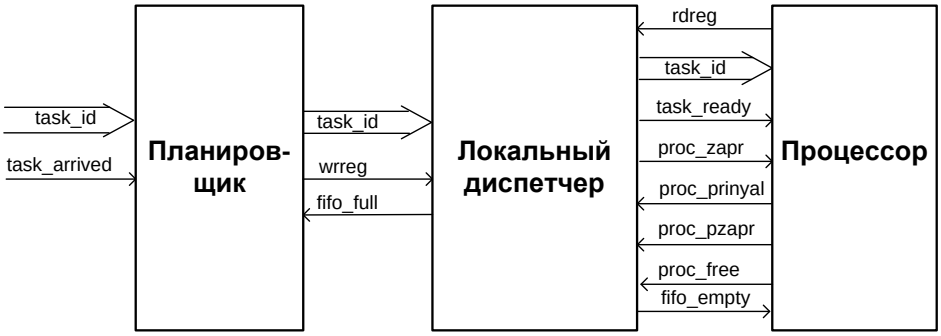


Рис. 2. Интерфейс взаимодействия локального диспетчера с планировщиком задач и процессором

Формализация алгоритма планирования задач

Алгоритм планирования/диспетчеризации основан на классической задаче «спящий парикмахер». Поступившая задача находится в планировщике до тех пор, пока не получит необходимые ей ресурсы, в число которых входит и место в очереди к одному из процессоров. Число мест в очереди ограничено. Если имеется свободное место в одной из очередей, то планировщик помещает новую задачу на это место. Принятая на обслуживание задача находится в очереди до тех пор, пока не поступит на выполнение в процессор. Если используется режим квантования, то незавершенная задача по окончании текущего кванта помещается в конец той же очереди, где она находилась ранее, иначе результат выдается пользователю, а в очереди освобождается одно место. При завершении задачи локальный диспетчер просматривает очередь, и если в ней имеются заявки на обслуживание, то назначается на выполнение процесс, стоящий в голове списка (очередь типа *FIFO*). Если очередь пуста, процессор переходит в режим ожидания [2].

Синхронизация процессов принятия задач в очередь и назначения очередной задачи процессору реализуется методом «рандеву» [3]. Формальное описание алгоритма взаимодействия процессов в данной задаче базируется на использовании моделей недетерминированных автоматов (НДА). Модели представляются в виде графа и/или систем канонических уравнений (СКУ), описывающих все реализуемые события управляющего алгоритма.

Основные частные события алгоритма представлены в табл. 1. В последней колонке показано их соответствие сигналам на схеме устройства управления процессами.

Обозначение события	Описание частного события	Сигналы на схеме (рис. 2)
S_I^t	Ожидание поступления новой задачи	*S_task_arrived*
S_U^{Qj}	Функция состояния очереди (количество задач в очереди *j*-го процессора)	*S_use [5…0]*
S_{FQ}^t	Все очереди заполнены	*S_fifo_full*
$\overline{S_{FQj}^t}$	В *j*-й локальной очереди имеются свободные места	$\overline{\text{S\_fifo\_fullj}}$
S_{SL}^{Qj}	Выбор очереди с минимальным числом задач и занесение новой задачи в выбранную очередь	*S_wrreg*
S_{Qj}^t	Задача в очереди *j*-го процессорного узла	*S_task_in_queue*
S_{ZPj}^t	Запрос *j*-го процессора локальный диспетчером	*S_proc_zapr_j*
S_{GPj}^t	Задача готова к обслуживанию в *j*-м процессоре	*S_task_ready_j*
S_{PZ}^{pj}	Подтверждение запроса *j*-м процессором	*S_proc_pzapr_j*
S_S^{pj}	Свободен *j*-й процессор	*S_proc_free*
S_{PT}^{pj}	Задачу *j*-й процессор принял	*S_proc_prinyal_j*
S_A^{pj}	Выполняет обслуживание задачи *j*-й процессор	Аппаратно не реализовано
S_O^{Qj}	Ожидание поступления задачи в очередь *j*-го процессора	Внутренний сигнал
S_{OF}^t	Удалить задачу из очереди	--------\\--------
S_{TO}^t	Снять задачу с исполнения	--------\\--------
S_E^{pj}	Выполнение задачи закончено	--------\\--------
S_{RT}^{pj}	Выдача результата выполнения текущей задачи	--------\\--------
S_{SZ}^{pj}	Процессор в режиме ожидания	--------\\--------

На основании словесного алгоритма планирования/диспетчеризации и введенных событий, система канонических уравнений будет иметь вид:

– для процесса «клиент», реализуемого локальным диспетчером до момента рандеву:

$$S_I^t(t+1) = x_Z \vee S_I^t \overline{S_{FQ}^t}\ ;\ S_O^{Qj}(t+1) = S_{FQ}^t \vee S_O^{Qj} \overline{S_{SL}^{Qj}}\ ;$$

$$S_{SL}^{Qj}(t+1) = \bigvee_{\forall j \in N} S_O^{Qj} S_U^{Qj}\ ;\ S_Q^t(t+1) = S_{SL}^{Qj} \vee S_Q^t \overline{S_{SZ}^{pj}}\ ; \qquad (1)$$

$$S_{ZPj}^t(t+1) = S_Q^t S_{SZ}^{pj} \vee S_{ZPj}^t \overline{S_{PZ}^{pj}}\ ;\ S_{GPj}^t(t+1) = S_{ZPj}^t S_{PZ}^{pj} \vee S_{GPj}^t \overline{S_{PT}^{pj}}\ ,$$

где x_Z означает наличие запроса задачи на постановку в очередь.

– для процесса «сервер», реализуемого процессором до момента рандеву:

$$S_{SZ}^{pj}(t+1) = S_S^{pj} \vee S_{SZ}^{pj} \overline{S_Q^t}\ ;\ S_{PZ}^{pj}(t+1) = S_{SZ}^{pj} S_Q^t \vee S_{PZ}^{pj} \overline{S_{ZPj}^t}\ ;$$

$$S_{PT}^{pj}(t+1) = S_{PZ}^{pj} S_{ZPj}^{t} \vee S_{PT}^{pj} \overline{S_{ZPj}^{t}}. \quad (2)$$

СКУ, описывающие события после рандеву:

$$S_A^{pj}(t+1)=S_{PT}^{pj} S_{GPj}^{t} \vee S_A^{pj} \overline{(S_E^{pj} \vee S_{PK}^{pj})};\ S_{TO}^{t}(t+1)=S_A^{pj} S_E^{pj} \vee S_{TO}^{t} \overline{S_{RT}^{pj}};$$

$$S_{RT}^{pj}(t+1)=S_A^{pj} S_E^{pj} \vee S_{RT}^{pj} \overline{S_{TO}^{t}};\ S_{OF}^{t}(t+1)=S_{RT}^{t} S_{TO}^{pj}; \quad (3)$$

$$S_S^{pj}(t+1)=S_{RT}^{t} S_{OF}^{pj}.$$

Следует иметь в виду, что выбор очереди для записи в нее вновь поступившей задачи выполняется в соответствии с выражением:

$$S_{SL}^{Qj} = \min_{(\forall j \in N)} \left\{ S_U^{Qj} \right\}. \quad (4)$$

Выражение (4) означает взятие функции минимального количества занятых мест из всего массива локальных очередей, где N – число процессоров (очередей) в многопроцессорной системе.

Структура устройства планирования с локальными очередями задач

Аппаратно устройство планирования представлено в виде трех типов блоков: блок назначения задач; N блоков очередей со схемами управления, где N – число процессоров в многопроцессорной системе; блок формирования очередей;

Блок очереди предназначен для хранения готовых к обслуживанию задач. Число этих блоков равно числу процессоров в системе. Запись в очередь производит диспетчер, по сигналам *S_wrreq,* подаваемым из блока планирования. Считывание содержимого регистра очереди производится по сигналу *S_rdreq*, поступающему от каждого процессора по индивидуальным линиям.

Входная шина данных формируется блоком планирования (*V_task_id_xz[7...0]*) и является общей для всех блоков очереди. Каждый процессор управляет только своей очередью, другими словами он может читать содержимое только из одного блока. Также каждый процессор постоянно сканирует свою очередь (семафорный сигнал *S_fifo_empty*) на предмет наличия идентификаторов задач. Если в регистрах имеется идентификатор хотя бы одной задачи, процессор читает его, если нет, то переходит в режим ожидания.

Выходная шина данных из каждого блока очереди заводится в индивидуальный процессор. Семафорный сигнал *S_fifo_full* используется блоком назначения задач для указания на заполнение очереди блока очереди. По этому сигналу диспетчер прекращает прием нового идентификатора в данную очередь.

Блок формирования очередей выбирает из всех N очередей блок очереди, в котором находится минимальное количество задач (содержащий наименьшее количество записей). Номер очереди с наименьшим числом записей передается в блок планирования по 3-х разрядной выходной шине *sel [2..0]*.

Блок назначения задач принимает новый идентификатор задачи и под действием сигналов *S_wrreq* помещает его в наименее заполненную очередь *FIFO*. Кроме того, на данный блок может возлагаться функция контроля состояния всех процессоров и функция перераспределения задач по очередям для выравнивания нагрузки на процессоры.

Моделирование алгоритма планирования задач на языке VHDL и анализ результатов

Для моделирования работы планировщика с локальными очередями была применена схема, показанная на рис. 3. Была смоделирована ситуация, когда в систему может поступить одновременно до восьми задач, причем любой процессор обслуживает одну задачу на протяжении случайного периода времени (от 6 до 11 тактов).

Частота некоторых запросов выбрана таким образом, чтобы возникали конфликты за доступ к процессору. Например, второй запрос xz[0] поступил в момент времени, когда процессор занят обслуживанием предыдущего запроса. В состоянии конфликта вновь поступившая задача устанавливается планировщиком в локальную очередь, время ожидания

в которой зависит от производительности процессора и трудоемкости задачи. Оценка таких задержек является предметом дальнейших исследований. Для этих целей авторами разрабатываются соответствующие модели массового обслуживания.

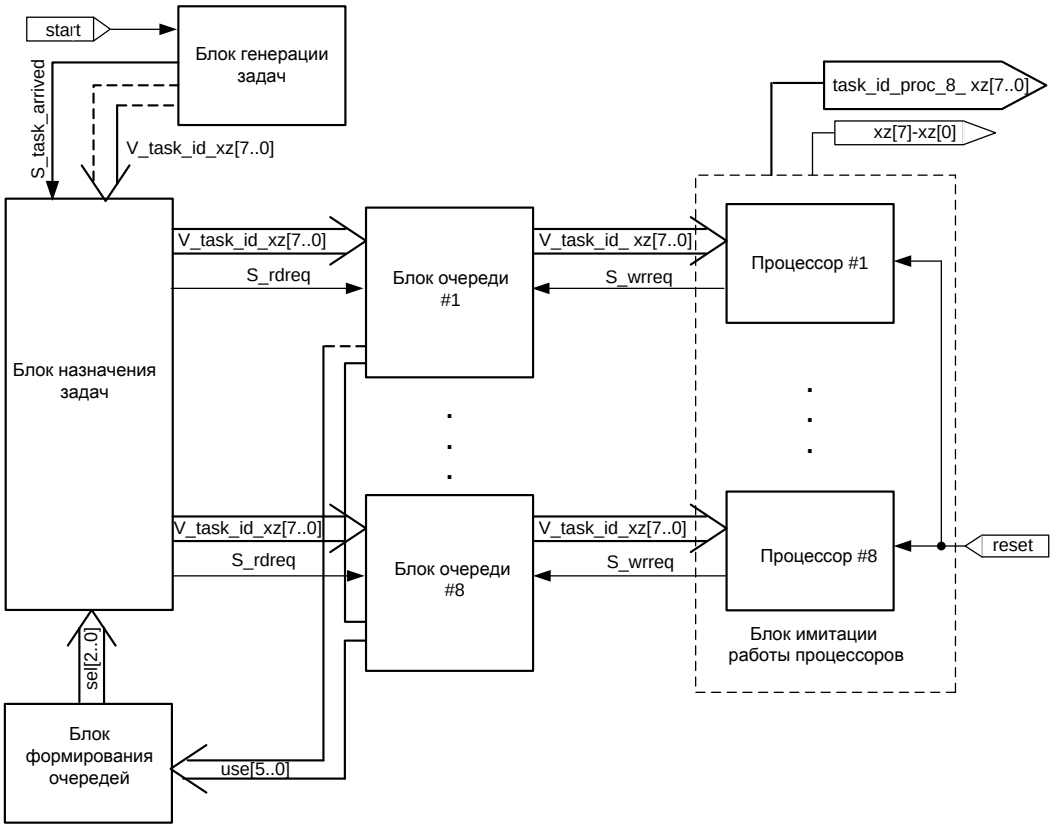


Рис. 3. Схема для проведения экспериментальных исследований

На рис. 4 приняты следующие обозначения: *xz*[7] – *xz*[0] – идентификаторы задач, поступающие на вход блоков очереди *FIFO*; *sa*[7] – *sa*[0] – обработка процессорами пользовательских задач.

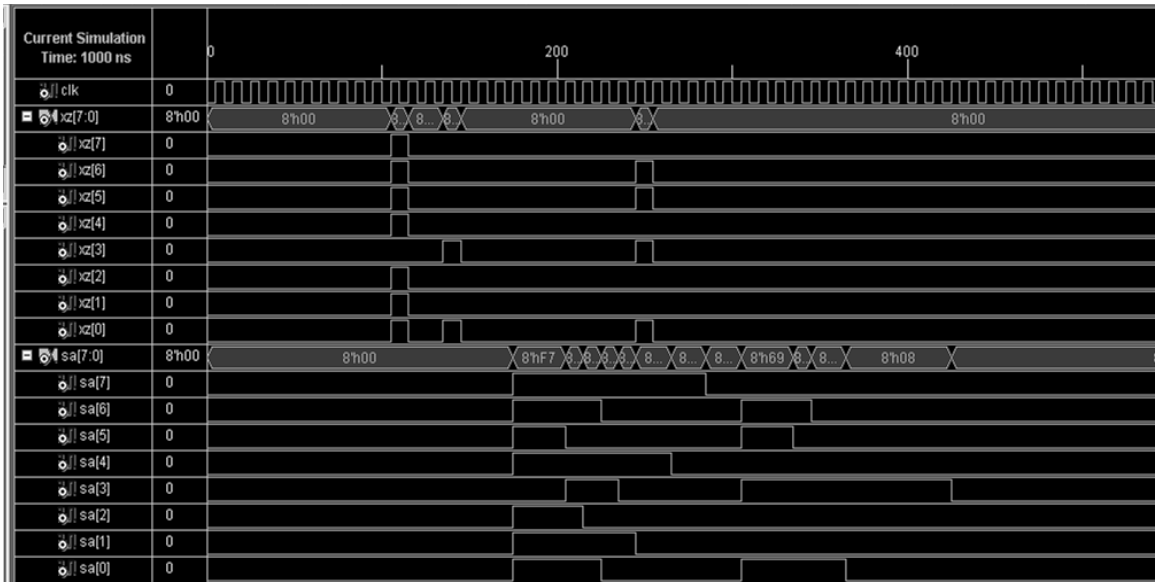


Рис. 4. Результаты моделирования

Из анализа временных диаграмм можно сделать следующие выводы. Задержка выполнения задач в условиях бесконфликтности (латентность) составляет всего 5–7 процессорных тактов. В современных ОСРВ, реализованных программно, задержка (обработка запроса) обычно составляет 5–10 тыс. процессорных тактов с учетом обработки прерывания по кванту.

Недостаток планировщика с локальными очередями заключается в отсутствии автоматической балансировки загрузки процессоров, что может приводить к неравномерности загрузки процессоров.

Библиографический список

1. Вашкевич, Н. П. Аппаратная поддержка диспетчера задач с глобальной очередью в многопроцессорных системах / Н. П. Вашкевич, Р. А. Бикташев, А. И. Меркурьев // Известия высших учебных заведений. Поволжский регион. Технические науки. – 2011. – № 3. – С. 3–14.

2. Вашкевич, Н. П. Недетерминированные автоматы и их использование для реализации систем параллельной обработки информации : моногр. / Н. П. Вашкевич, Р. А. Бикташев. – Пенза : Изд-во ПГУ, 2016. – 394 с.

3. Вашкевич, Н. П. Формализация алгоритмов управления параллельными процессами на основе использования механизма рандеву / Н. П. Вашкевич, Р. А. Бикташев // XXI век: итоги прошлого и проблемы настоящего плюс. – 2011. – № 3 (03). – С. 115–124.

Р. А. Бикташев

Пензенский государственный университет, Пенза, Россия

МОДЕЛЬ УПРАВЛЕНИЯ ПРОЦЕССАМИ В МНОГОПРОЦЕССОРНЫХ СИСТЕМАХ НА ОСНОВЕ СТОХАСТИЧЕСКИХ СЕТЕЙ МАССОВОГО ОБСЛУЖИВАНИЯ

Представлена аналитическая модель для оценки показателей производительности функций планирования и диспетчеризации многопроцессорных операционных систем.

В многопроцессорных системах (МПС) существует проблема деградации производительности вызываемой трудоемкими функциями управления процессами. Эти функции выполняют планировщики/диспетчеры задач, которые в МПС запрашиваются процессорами гораздо чаще, чем в однопроцессорных системах. При проектировании специализированных многопроцессорных операционных систем необходимо прогнозировать показатели производительности, которые чаще всего выражается через время ответа системы на запрос.

Планирование зачастую предполагает использование глобальной очереди готовых задач, которая представлена в многопроцессорной системе (МПС) в виде таблицы процессов, хранящейся в общей памяти. При переключениях процессов выбираемая из очереди задача поступает на обработку в любой свободный процессор, причем не обязательно в тот, в котором она выполнялась ранее. Поэтому в таких МПС существует явление перезагрузки кэш, когда прерванная задача с высокой вероятностью может быть направлена для продолжения обслуживания в другой процессор. Названное явление увеличивает частоту кэш-промахов и неизбежно приводит к снижению производительности многопроцессорной системы. Кроме того велика вероятность возникновения конфликтов, когда множество свободных процессоров состязаются между собой за доступ к единственной очереди задач. На разрешение конфликтов затрачивается процессорное время, что увеличивает потери производительности процессоров МПС.

Пусть МПС состоит из n статистически идентичных процессоров, планировщика, основная функция которого формирование очереди готовых задач и диспетчера, переключающего процессы (потоки). Завершенные задачи снимаются планировщиком с исполнения, поэтому считается, что планировщик выступает в качестве источника и поглотителя заявок.

Задача, заставшая все процессоры занятыми обслуживанием, ставится в очередь, иначе поступает на обслуживание в любой свободный процессор. Переключение контекста производит диспетчер, в результате чего текущая задача, если ее выполнение не закончилось, направляется в очередь, на то место, которое определяется алгоритмом планирования. Поскольку заявки назначаются на любой свободный процессор, то процессорный блок представляется в виде многоканальной СМО.

Диспетчер являются общим (разделяемым) ресурсом для всех запрашивающих процессоров, поэтому модель диспетчера представлена в виде одноканальной СМО.

Структура сетевой модели многопроцессорной системы с планировщиком/ диспетчером задач на основе глобальной очереди представлена на рис. 1. В модели принято, что запросы задач на обслуживание генерируются планировщиком с интенсивностью λ_0. Заявки назначаются в любой свободный процессоров и выполняются в течение случайного времени со средним значением $v_{ср} = 1/\mu_{ср}$, где $\mu_{ср}$ – интенсивность обслуживания в каждом из n процессоров.

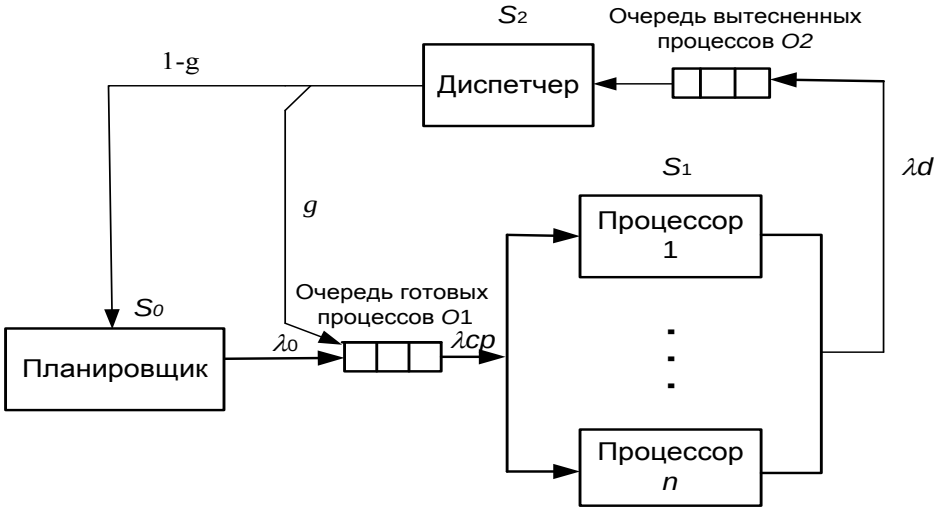


Рис. 1. Схема модели планировщика с общей очередью задач в составе многопроцессорной системы

Таким образом, обслуживание в процессорном блоке (S_1) в соответствии с нотацией Кендалла [1] представляется в виде многоканальной СМО типа $M/M/n/m$, с числом обслуживающих каналов n и ограничением длины очереди O_1 числом мест m.

Процессорный блок формирует поток запросов λ_d к диспетчеру на переключение процесса. Если диспетчер свободен, то прерывает выполнение текущей задачи в i-м процессоре, и с вероятностью g ставит ее в очередь O_1, (при наличии свободных мест). Далее диспетчер назначает i -му процессору новую задачу из очереди O_1, если она не пуста.

Если диспетчер занят, то процесс ставится в очередь (приостанавливается). В реальной системе приостановка осуществляться двумя вариантами: путем перевода процессора в спящий режим, либо путем выполнения процессором холостых команд в цикле (спиннинг).

Задача, получившая полное обслуживание, с вероятностью $(1–g)$ удаляется из сети после последнего переключения контекста, выполняемого диспетчером. Удаление проис-

ходит перемещением ее в фиктивную СМО S_0. На обслуживание в диспетчере затрачивается среднее время $v_d = 1/\mu_d$, где μ_d – интенсивность обслуживания.

Исходя из условия баланса, которое заключается в равенстве средних значений входящих и выходящих потоков для каждого узла сети, интенсивность потока задач, поступающих в очередь O_2, составляет $\lambda_{cp} = \lambda_d = \lambda_0 + g\lambda_d$.

При переключениях контекста текущая задача помещается в очередь в соответствии с алгоритмом планирования. Когда прерванной задаче очередной раз будет предоставлен процессорное время, она с вероятностью *f* будет направлена в процессор, в котором она обслуживалась ранее, а с вероятностью (1– *f*) в другой свободный процессор. Тогда с вероятностью (1– *f*) произойдет перезагрузка кэш. Если принять равновероятное распределение задач по процессорам, то *f* = 1/*n*. Если через t_c обозначить время, необходимое для перезагрузки кэш (случайная величина), а среднее время (математическое ожидание) выполнения задачи между двумя переключениями контекста (случайная величина) в процессоре – через t_t, то время обслуживания в *i*-м процессоре составит $v_{\text{ср}} = t_t + ft_c = t_t + t_c / n$.

Время обслуживания в диспетчере составит $v_d = (t_d + \tau)$, где t_d – среднее время работы программы диспетчеризации, τ – среднее время, необходимое для переключения контекста. Под временем работы программы диспетчеризации t_d, в данном случае, понимается время, затрачиваемое на выполнение системного вызова, вычисление приоритета процесса, на перераспределение очереди готовых процессов в таблице процессов, на передачу процессору адреса (идентификатора) нового процесса.

При переключениях контекста по кванту время обслуживания в *i*-м процессоре составит [2] $v_{\text{ср}} = (t_q + t_c)$, где t_c – время, необходимое для перезагрузки кэш (случайная величина); t_q – длительность кванта (постоянная величина). Время обслуживания одного процесса в диспетчере определяется выражением $v_d = (t_d + \tau)$.

Модель процессорного блока. Рассмотрим модель процессорного блока, представленная в стохастической сети многоканальной СМО S_1. СМО имеет единственную очередь задач типа *FIFO*, т.е. обслуживание происходит в порядке их поступления. Задачи, не успевшие выполниться за отведенный этап обслуживания между двумя переключениями контекста или за квант процессорного времени, вновь устанавливаются в ту же очередь на дообслуживание. Задача, поступившая в момент, когда все каналы (процессоры) заняты, становится в очередь и ожидает обслуживания. Число задач (процессов), которые одновременно выполняются в процессорном блоке $z = \lambda_{\text{ср}} / \mu_{\text{ср}}$, и может быть представлено как среднее число процессоров, занятых обслуживанием. Очередь имеет ограниченное число мест, поэтому если она заполнена, то новые задачи не принимаются на обслуживание. Этот факт моделируется потерей заявок, однако эта особенность на характеристики СМО не влияет. Обозначим число мест в очереди через *m*, тогда СМО S_1 не может вместить более *n* + *m* задач, из которых *n* находится на обслуживании, а *m* ожидают.

Определив параметры задач и параметры устройств достаточно легко можно найти единичное время пребывания задачи в процессорном блоке $u_{\text{ср}}$ [1].

Модель диспетчера. Заявки на обслуживание в диспетчере формируются процессами, выполняющимися в процессорном блоке. Если в момент поступления заявки диспетчер занят, то выполнение *i*-го процесса приостанавливается, причем приостановка моделируется очередью O_2. Если диспетчер свободен, он обслуживает задачу, на что уходит среднее время v_d.

Модель диспетчера представляется замкнутой одноканальной СМО (S_2) с ограниченным числом источников заявок на обслуживание, что по классификации Кендала представляется как *M/G*/1/*n*.

Выше определено, что в многоканальной СМО, моделирующей обслуживание в процессорном блоке, занято обслуживанием в среднем z из общего числа *n* процессоров, где z – число запущенных задач, т.е. задач, получающих процессорное время (квант про-

цессорного времени). Следовательно, z процессов (процессоров, которые обслуживают указанные процессы) являются источниками заявок для обслуживания диспетчером задач (СМО S_2). Если диспетчер свободен, то он принимает задачу (процесс) на обработку, если занят, то задача ставится в очередь. В реальной МПС очередь состоит из приостановленных процессов, которые обратились к диспетчеру задач в момент его занятости.

Пусть $\bar{z}$ – среднее число приостановленных задач (процессоров) из общего числа работающих z, тогда $(z-\bar{z})$ процессов являются источником заявок к диспетчеру, средняя интенсивность которого составляет $\lambda_d = (z-\bar{z})\cdot\lambda_1$, где λ_1 интенсивность потока задач, формируемым одним процессором. Вследствие баланса потоков на входе и выходе СМО S_2, получим: $\left(z-\overline{z}\right)\cdot\lambda_{ср} = (1-\pi_0)\cdot\mu_d$, где μ_d – интенсивность обслуживания задач диспетчером, π_0 – вероятность отсутствия запросов к СМО S_2.

Зная параметры диспетчера задач можно легко определить единичное время пребывания u_d заявки в СМО S_2 по формулам, приведенным в [1].

С учетом коэффициентов передачи сети $\varphi_{ср} = \eta_{ср} / \lambda_0$ и $\varphi_d = \lambda_d / \lambda_0$ время пребывания заявки в сети время ответа системы () определяется суммой времен пребывания заявки в процессорном блоке и в диспетчере $U = \varphi_{ср}\cdot u_{ср} + \varphi_d \cdot u_d$, где $u_{ср}$, u_d – единичные времена ответа СМО S_1 и S_2 соответственно [2]. Последнее выражение является временем ответа системы.

Библиографический список

1. Таха, Х. А. Введение в исследование операций / Х. А. Таха. – М. : Вильямс, 2005. – 912 с.

2. Бикташев, Р. А. Математическое моделирование функций управления взаимодействующими процессами и ресурсами многопроцессорных систем / Р. А. Бикташев, А. И. Мартышкин, Н. Г. Востоков // Оптико-электронные приборы и устройства в системах распознавания образов, обработки изображений и символьной информации. Распознавание – 2013 : сб. ст. XI Междунар. науч.-техн. конф. – Курск : ЮЗГТУ, 2013. – С. 218–220.

Р. Н. Федюнин

Пензенский государственный университет, Пенза, Россия

СПЕЦИАЛИЗИРОВАННЫЕ АРИФМЕТИКО-ЛОГИЧЕСКИЕ УСТРОЙСТВА НА БАЗЕ КОНВЕЙЕРНЫХ МАТРИЧНЫХ ВЫЧИСЛИТЕЛЕЙ

Итеративно-битовые процессоры [1, 2] ориентированы на реализацию двуместных арифметических операций и отношений равенства и неравенства значений данных относительно некоторого критерия. В ряде случаев функциональные возможности процессорных элементов обеспечивают также возможность выполнения и операций преобразования структур данных – операций сжатия двоичных векторов, формирования векторов-масок и т.д.

Концептуально организация арифметико-логического блока процессоров на базе итеративно-битовых процессорных пространств определяется функциональными возможностями процессорных модулей многомерной вычислительной среды, топологией межпроцессорных связей, способами управления настройкой и вычислениями, количеством каналов и способами ввода и вывода информации из вычислительного пространства. В связи

с этим общая организация АЛУ процессора представима структурой, представленной на рис. 1.

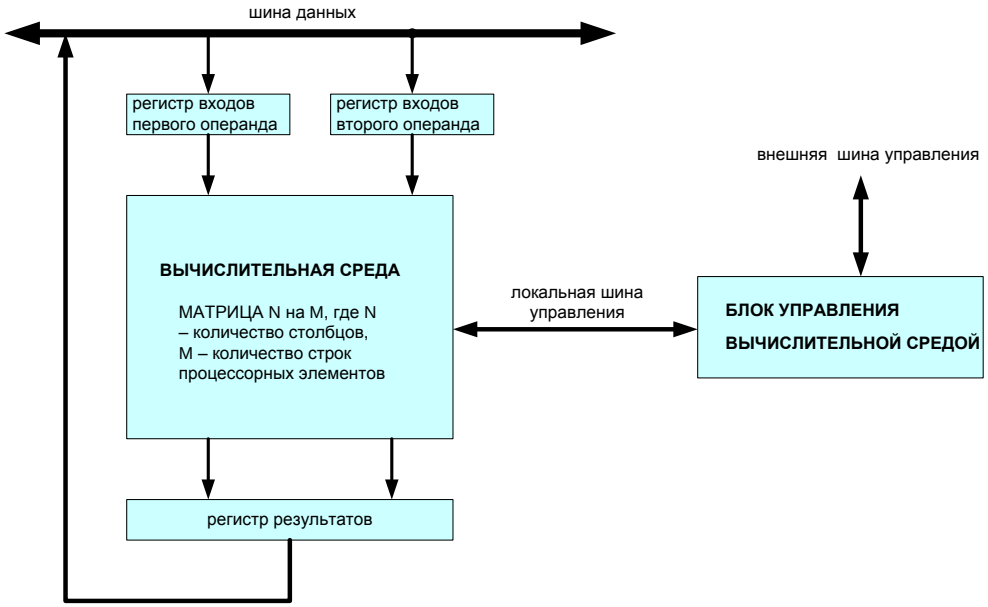


Рис. 1. Организация АЛУ итеративно-битового процессора

Вычислительная среда или как ее еще называют однородная вычислительная среда (ОВС) (рис. 1) – является операционной частью процессора и организована как матрица процессорных элементов (ПЭ). Иллюстрация организации операционной части процессора приведена на рис. 2, где 1,1–1, *m*, 2,1–2, *m*, …, *n*,1–*n*,*m* – где ячейки процессорных элементов, p_0–p_n, b_0–b_m – информационные входы, c_0–c_n, s_0–s_m – информационные выходы, PRG – управляющий вход, соединенный с управляющими входами prg, каждого процессорного элемента.

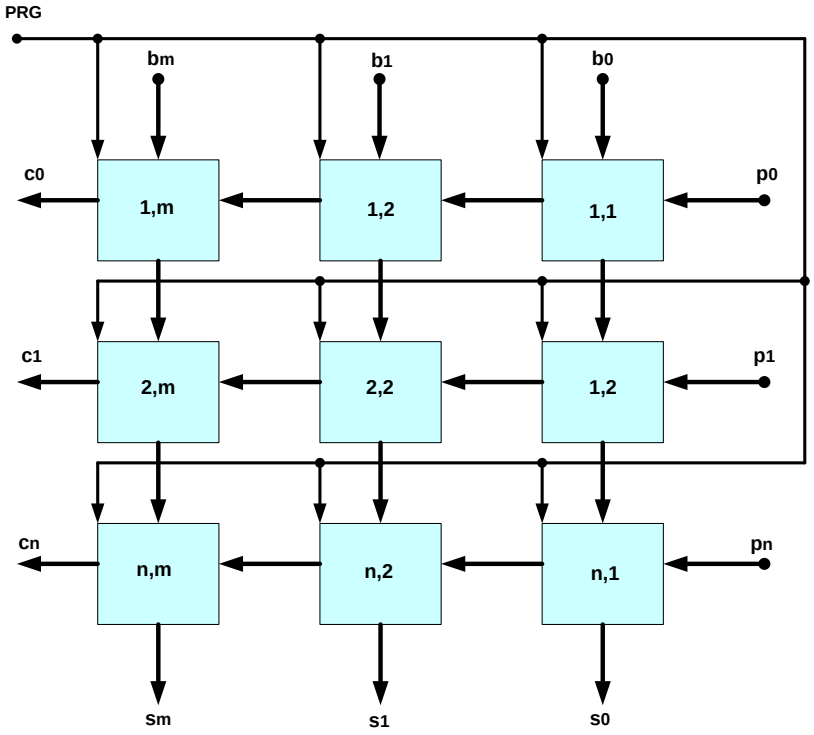


Рис. 2. Организация операционной части АЛУ процессора

Анализ методов организации вычислений в многомерных арифметико-логических процессорах показывает, что для выполнения массивных операций наиболее эффективна топология информационных межпроцессорных каналов типа «решетка».

Такая топология межпроцессорных связей обеспечивает возможность реализации арифметико-логических операций на базе простого одноразрядного процессорного модуля.

В качестве процессорного модуля вычислительного пространства для выполнения арифметико-логических вычислений над данными, представляемыми n-разрядными двоичными векторами, достаточно эффективен процессорный элемент (ПЭ) [3, 4], организация которого приведена на рис. 3.

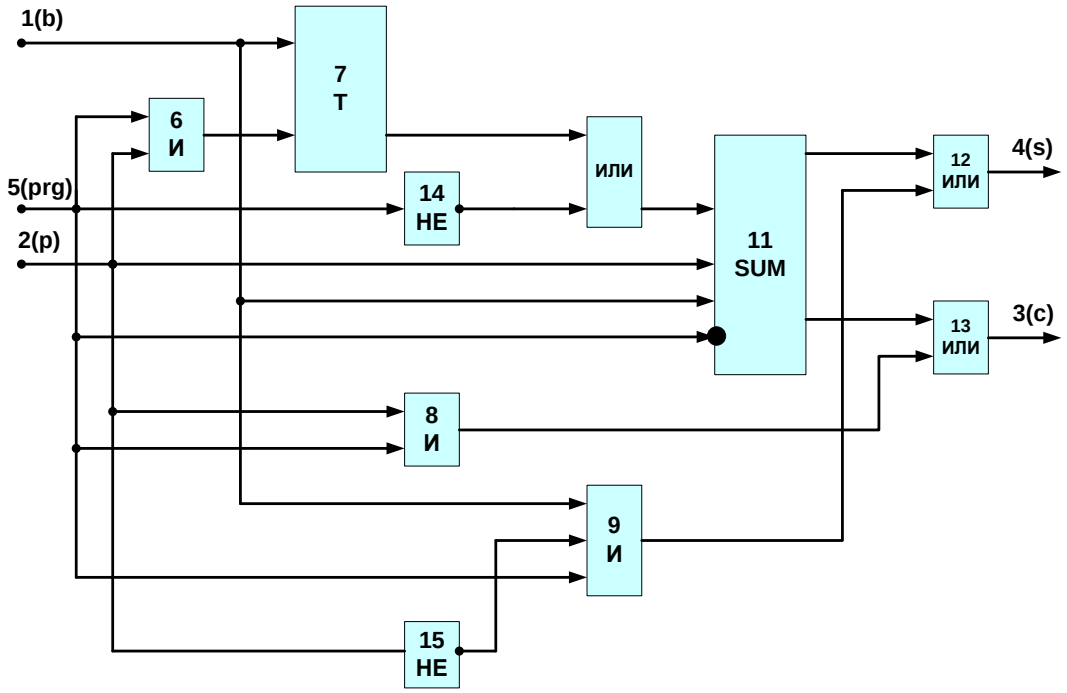


Рис. 3. Организация процессорного элемента АЛУ итеративно-битового процессора

Процессорный элемент реализует следующую систему логических функций:

$$
\begin{aligned}
Q(t) &= prg \cdot b \cdot p \\
s &= \overline{prg} \cdot b \cdot p \cdot Q(t) \vee prg \cdot \overline{p} \cdot b \\
c &= \overline{prg} \cdot b \cdot p \cdot Q(t) \vee prg \cdot p
\end{aligned}
\qquad (1)
$$

где **b** и **p** – сигналы, соответственно подаваемые на информационные входы 1 и 2 ячейки;

prg – сигнал, формируемый соответственно на управляющем входе 5;

c и **s** – сигналы, формируемые соответственно на информационных выходах 3 и 4 ячейки;

Q(t) – состояние триггера 7 в момент времени *t.*

Однородная вычислительная структура позволяет выполнять построчную загрузку кода с группы информационных входов b_0–b_m в триггеры ячеек однородной структуры и реализовывать режим вычисления.

Загрузка кода, для реализации арифметической операции, осуществляется с группы входов b_0, b_1,..., b_m ОВС в триггеры ПЭ соответствующей строки по каналам коммутации b_i–s_i.

Настройка ячеек ОВС на работу в режиме загрузки кода осуществляется подачей исходного вектора на группу входов b_0, b_1, ..., b_m ОВС, при этом младший разряд исходного вектора b_0 подается на первый информационный вход первой ячейки первого столбца, первой строки (на вход b_0 ОВС), при этом на управляющий вход PRG однородной структуры подается сигнал логической единицы, а на группу входов p_0, p_1, ..., p_n ОВС подается управ-

ляющий вектор, единичный бит которого указывает строку, в которую необходимо загрузить код вектора данных. В результате вектор с группы входов b_0, b_1, ..., b_m ОВС фиксируется в триггерах ПЭ нужной строки. Заметим, что загрузка кода в ячейки ОВС осуществляется в следующем порядке: первыми загружаются ячейки самой последней строки, затем предпоследней и т.д., последними будут загружены ячейки первой строки.

ПЭ в данном режиме реализует следующую систему логических функций:

$$\begin{aligned} s &= prg \cdot \overline{p} \cdot b \\ c &= prg \cdot p \\ Q(t) &= prg \cdot p \cdot b \end{aligned} \qquad (2)$$

Пример работы однородной вычислительной структуры в режиме загрузки кода приведен на рис. 4. На рис. 4 – показана загрузка кода 100_2 в ячейки самой последней строки. Для чего на группу входов p_0-p_2 подается управляющий вектор 100_2 , в результате чего ячейки последней строки настраиваются на выполнение операции загрузки кода, а ячейки всех остальных строк – на выполнение операции трансляции кода, при подаче на управляющий вход PRG ОВС активного сигнала, равного уровню логической единицы вектор 100_2 с группы входов b_0, b_1, b_2 загружается в ячейки самой последней строки.

Затем, по аналогии, на рис. 4 – иллюстрируется загрузка кода 0112 в ячейки третьей строки, для чего на группу входов p0-p2 подается управляющий вектор 0102 а на управляющий вход PRG ОВС подается активный сигнал, равный уровню логической единицы. Далее во вторую строку ОВС загружается битовый вектор 0112, для чего на группу входов p0-p2 подается управляющий вектор 0102 а на управляющий вход PRG ОВС подается активный сигнал, равный уровню логической единицы, а в первую строку загружается вектор – 0102, при подаче на группу входов p0-p2 управляющего вектора 0012 а на управляющий вход PRG ОВС активного сигнала, равного уровню логической единицы В данном примере видно, что заполнение ПЭ кодами векторов осуществляется построчно и снизу вверх, т.е. первый вектор загружается в самую последнюю строку однородной структуры, и т.д. до полного заполнения структуры или исчерпания векторных массивов.

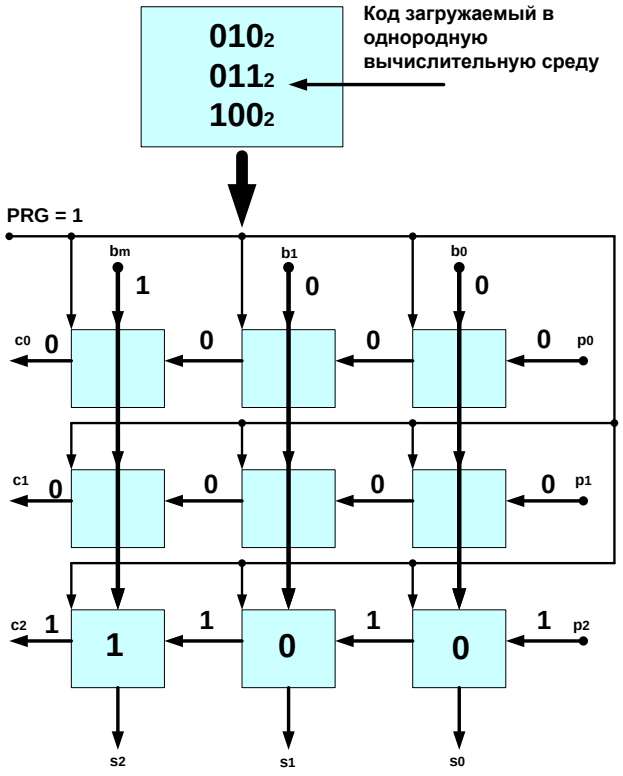


Рис. 4. Загрузка данных и процесс вычисления

После реализации подготовительного режима, устройство выполняет функцию сумматора, складывая вектора, загруженные в ПЭ ОВС в режиме загрузки.

Настройка ячеек ОВС на работу в режиме вычисления осуществляется подачей на управляющий вход PRG однородной структуры сигнала логического нуля. В результате чего в каждой ЯОС ОВС включаются цепи суммирования и реализуется следующая система логических функций:

$$\begin{aligned} s &= b \cdot p \cdot Q(t), \\ c &= b \cdot p \cdot Q(t), \end{aligned} \tag{3}$$

т.е. в каждой ячейке ОВС происходит сложение данных, хранящихся в триггере данной ячейки, с данными, поступающими в данную ячейку по входам **b** и **p**. Результат операции фиксируется на группе входов s_0–s_m.

Пример работы однородной вычислительной структуры в режиме вычислений приведен на рис. 5. На все информационные входы ПЭ ОВС подается сигнал логического нуля. В данном примере предполагается, что загрузка в ячейки ОВС необходимых для сложения векторов уже выполнена. Для начала выполнения операции сложения, на управляющий вход PRG подается сигнал равный нулю, в результате каждая ЯОС ОВС выполняет систему функций 4. Результат формируется по столбцам, и из примера видно, что если в столбце четное количество единиц то результат суммы столбца 0, иначе, если в столбце нечетное число единиц, то результат суммы столбца 1, при этом также учитывается формирование переносов из младших в старшие разряды числа. Результат вычислений фиксируется на группе информационных выходов s_0–s_6.

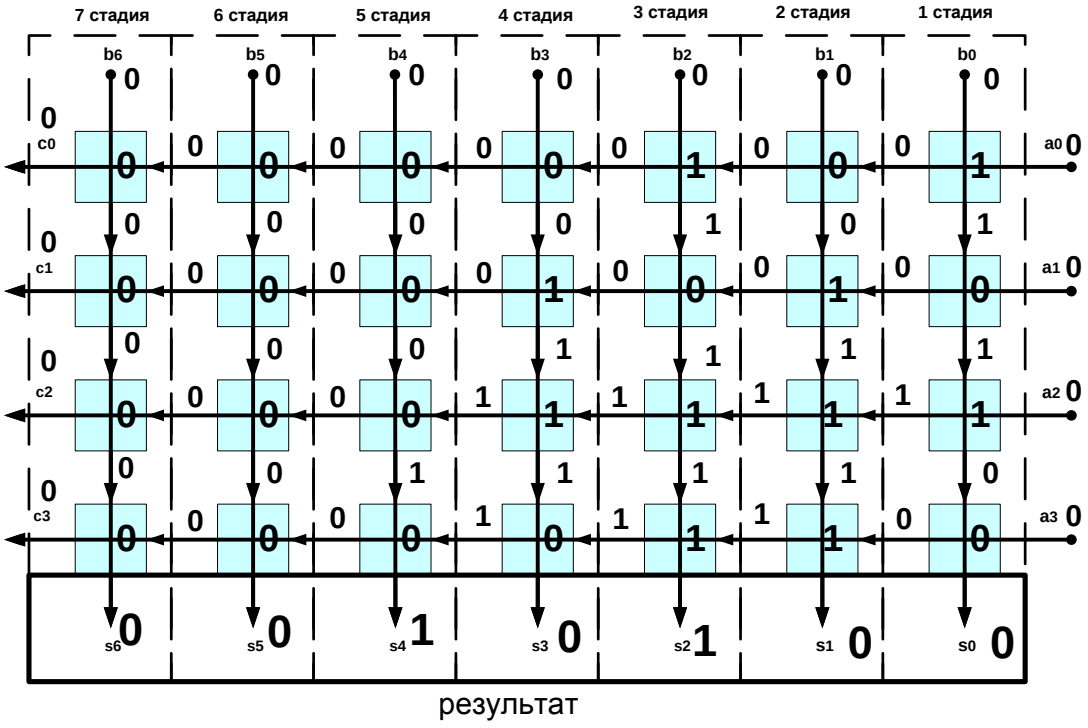


Рис. 5. Формирование результата

Таким образом, рассмотрев частный пример построения однородной вычислительной структуры, можно сделать следующие выводы.

Итеративно-битовая архитектура вычислительного устройства является решением, агрессивно эксплуатирующим параллелизм «конвейерного» типа. Несмотря на принципиальную дороговизну решения (применение специализированных аппаратных решений), систолы (ПЭ) являются одним из самых эффективных методов решения специализированных задач, так как часто обеспечивают на конкретной задаче быстродействие, недостижимое

для большинства микропроцессорных архитектур. Значительное удешевление программируемых логических матриц [5, 6] сводит на нет экономическую составляющую вопроса применения и пробуждает интерес исследователей к применению ОВС в сложных вычислительных задачах.

Итеративно-битовые системы являются очень специализированными вычислителями и производятся в основном под конкретную задачу. Фактически, задача построения вычислителя сводится к построению аппаратного конвейера, имеющего достаточно большое время получения результата (т.е. большое количество ступеней) но при этом сравнительно маленькое время между последовательной выдачей результатов, так как значительное количество промежуточных значений обрабатывается на разных ступенях конвейера.

Библиографический список

1. Фрумкин, А. И. Систолические вычисления / А. И. Фрумкин. – М. : Высш. шк., 1988. – 457 с.
2. Федюнин, Р. Н. Устройства конвейерно-параллельной обработки информации / Р. Н. Федюнин. – М. : Научное обозрение, 2005. – С. 72–80.
3. Пат. РФ № 2004136518/09 Ячейка однородной вычислительной среды / Федюнин Р. Н. Князьков В. С. ; от 12 мая 2006 г.
4. Пат. РФ № 2004136518/09 Ячейка однородной среды / Князьков В. С., Федюнин Р. Н. ; от 15 мая 2006 г.
5. URL: www.xilinx.com
6. URL: www.altera.com

И. А. Беззатеев[1], В. Н. Дубинин[1], Д. А. Уваров[1], В. В. Вяткин[2]

[1] Пензенский государственный университет, Пенза, Россия
[2] Технический университет Aalto University, г. Хельсинки, Финляндия

РЕАЛИЗАЦИЯ РАСПРЕДЕЛЕННЫХ СЕМАФОРОВ В АРХИТЕКТУРЕ IEC 61499 НА ОСНОВЕ АЛГОРИТМОВ ДЛЯ РЕШЕНИЯ ЗАДАЧ КОНСЕНСУСА В СЕТИ НЕНАДЕЖНЫХ ПРОЦЕССОВ

Предложен подход к реализации распределенных семафоров в архитектуре IEC 61499 на основе алгоритмов для решения задач консенсуса в сети ненадежных процессов, что позволяет разрабатывать управляющие функционально-блочные приложения со сложными видами взаимодействий, основанных на разделении общих ресурсов.

Международный стандарт IEC 61499 определяет концепцию и системную модель для построения распределенных управляющих и измерительных систем нового поколения [1]. Для взаимодействия между ресурсами и устройствами системы в данном стандарте используется механизм передачи сообщений посредством коммуникационных функциональных блоков (ФБ) CLIENT/SERVER и PUBLISH/SUBSCRIBE. Данный механизм является единственным и довольно низкоуровневым, что не позволят напрямую строить распределенные системы со сложными видами взаимодействий, включая критические секции, «производитель-потребитель», «читатели-писатели» и т.д., которые во многом ориентированы на взаимодействие через общую память и основаны на использовании семафоров [2]. Простой, но в то же время показательный пример использования семафора в управляющих системах – это взаимоисключающая работа нескольких ресурсов (или устройств) IEC

61499 с общим технологическим оборудованием, данными или коммуникационной сетью.

Можно выделить два подхода к реализации семафоров в распределенной системе – централизованный и децентрализованный. В работе [3] была предложена клиент-серверная реализация семафора в архитектуре IEC 61499. Однако, данная реализация, где сам семафор реализован как сервер, фактически является централизованным решением, имеющим недостатки, присущие всем централизованным системам – низкую надежность и производительность, плохую масштабируемость.

В последнее время все большее внимание исследователей привлекают методы групповых взаимодействий в распределенных системах, в частности алгоритмы (протоколы) достижения консенсуса между компонентами распределенной сети – Paxos [4] и Raft [5]. Семейство протоколов Paxos используется в распределенной службе блокировок Chubby компании Google, используемой, в свою очередь, в BigTable (проприетарная высокопроизводительная база данных). Microsoft использует Paxos в своей системе автоматического управления кластерами. Алгоритм Raft относительно молод, но уже находит применение в крупных IT-проектах. Facebook использует NoSQL базу данных HydraBase в механизме сообщений, в основе которой лежит кластер, поверх которого развернут Raft.

Основываясь на факте, что семафор это фактически разделяемая булева или целочисленная переменная с определенными на ней операциями, можно предположить, что протоколы Paxos и Raft могут вполне успешно использоваться при реализации распределенных семафоров. Немногочисленные публикации подтверждают это (например, [6]).

Протоколы семейства Paxos гарантируют три следующих показателя: нетривиальность, консистентность и живучесть. При этом рассматриваются варианты решения задачи консенсуса в зависимости от количества вычислителей, количества задержек для получения результата, активности участников, количества отправленных сообщений и типов отказов. В протоколе Paxos узлы распределенной системы имеют следующие роли: Client, Proposer, Acceptor, Learner. Базовый алгоритм Paxos состоит из следующих этапов: Prepare («предложение»), Promise («обещание»), Accept («принять»), Accepted («принято»).

Протокол Raft разрабатывался с учетом недостатков предыдущего протокола Paxos. При выборе ключевых идей, предпочтение отдавалось более простым и практичным решениям. Тем не менее, несмотря на относительную простоту, Raft обеспечивает безопасную и эффективную реализацию машины состояний поверх кластерной вычислительной системы. Каждый узел кластера Raft имитирует копию конечного автомата, а цель протокола – поддерживать согласованность между копиями. Репликация позволяет системе продолжать обслуживать клиентов всякий раз, когда доступно большинство машин. Однако сохранение согласованности между репликами затруднено при наличии асинхронности, сбоев сети (падений пакетов, дублирования и переупорядочения) и сбоев узлов (сбоев и перезагрузок). К особенностям Raft относятся: 1) четкое разделение фаз; 2) наличие явно выделенного лидера; 3) возможность динамического изменения размера кластера.

В данной работе разработана библиотека функциональных блоков (ФБ) для реализации алгоритмов достижения консенсуса Paxos и Raft, позволяющая существенно расширить возможности проектирования распределенных управляющих приложений со сложными видами взаимодействий. Данная библиотека реализована в системе NxtStudio [7]. Ввиду того, что функционально-блочная реализация протоколов Paxos и Raft довольно объемная, описание этой реализации приводится лишь в ограниченном объеме.

Функционально-блочная реализация системы «Распределенный семафор на основе протокола Paxos» включает 14 ФБ, разделенных на следующие группы: 1) составные ФБ (CLIENT), реализующие функции по генерации клиентом запросов на взятие/освобождение семафора; 2) функциональные CAT-блоки для организации человеко-машинного интерфейса; 3) составные ФБ (PROPOSER, ACCEPTOR, LEARNER), реализующие соответствующие роли протокола Paxos. Составной ФБ PROPOSER включает сеть ФБ, представленную на рис. 1.

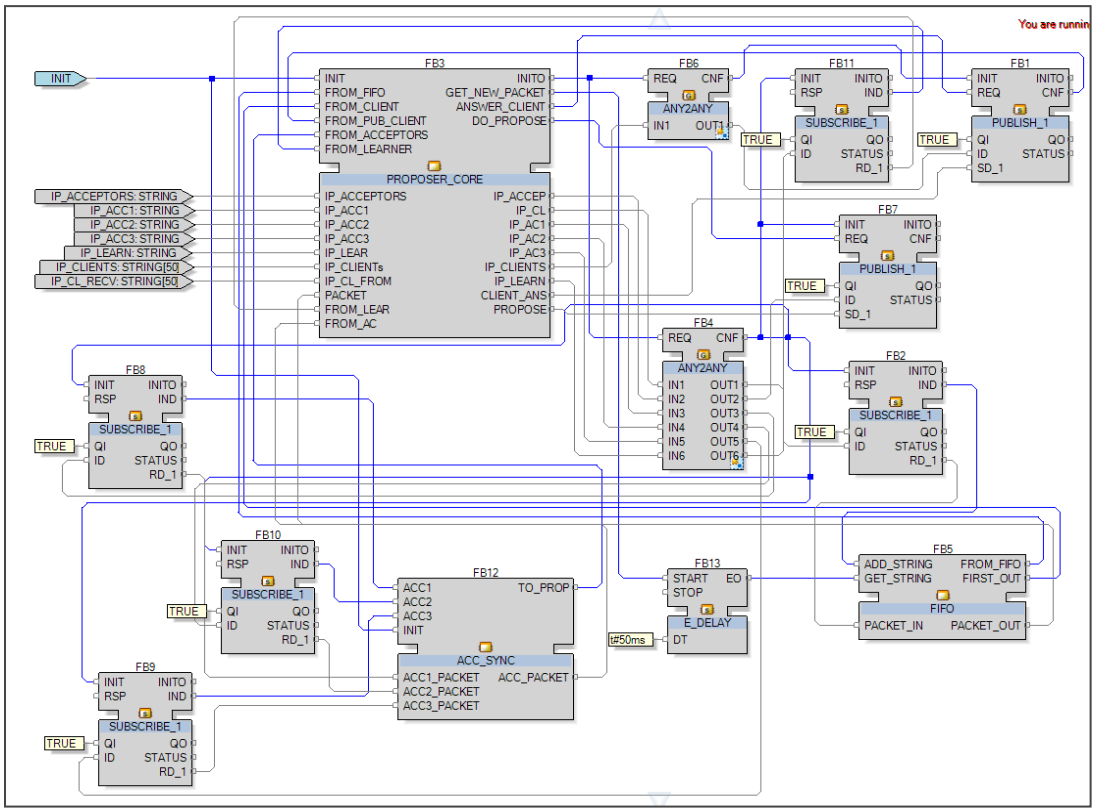


Рис. 1. Структура составного ФБ PROPOSER

Спроектированная в среде NxtStudio система «Распределенный семафор на основе Paxos» корректно функционирует при обработке запросов на захват/освобождение семафора от нескольких клиентов. При полном отказе модуля Acceptor система сохраняет свою работоспособность за счет особенностей протокола Paxos, если при этом сохраняется кворум. Описанные результаты подтверждены путем тестирования проекта на эмуляторе SoftPLC, входящим в среду исполнения NxtStudio.

Функционально-блочная реализация системы «Распределенный семафор на основе протокола Raft» включает 20 ФБ, разделенных на следующие группы: 1) составные ФБ (CLIENT), реализующие функции по генерации клиентом запросов на взятие/освобождение семафора; 2) функциональные CAT-блоки для организации человеко-машинного интерфейса; 3) составные ФБ (RAFT), реализующие функционал модулей протокола Raft.

Структура составного ФБ RAFT представлена на рис. 2.

Реализованная в среде NxtStudio система «Распределенный семафор на основе Raft» корректно функционирует при обработке запросов на захват/освобождение семафора от нескольких клиентов. При полном отказе нескольких Raft-модулей (для функционирования системы требуется минимум два модуля) система сохраняет свою работоспособность за счет заложенных в алгоритм Raft репликации данных и алгоритма выбора нового лидера. Описанные результаты подтверждены путем тестирования проекта на эмуляторе SoftPLC.

Ниже приведено краткое сравнение реализаций Paxos и Raft. Протокол Paxos подразумевает наличие как минимум трех ролей с различным функционалом, в то время как в алгоритме Raft количество ролей сведено к одной, которая функционирует в двух режимах. Для сохранения работоспособности в системе на основе Paxos необходимо наличие одного модуля Proposer, двух модулей Acceptor и одного модуля Learner. Система на основе Raft функционирует при наличии одного модуля Raft, работающего в режиме лидера, и одного модуля Raft, работающего в режиме подчиненного. Таким образом, итоговая сеть ФБ на основе Paxos, получается более громоздкой, чем аналогичная сеть на основе Raft. В результате разработки системы в среде NxtStudio весь функционал протокола Raft вместился в составной ФБ RAFT.

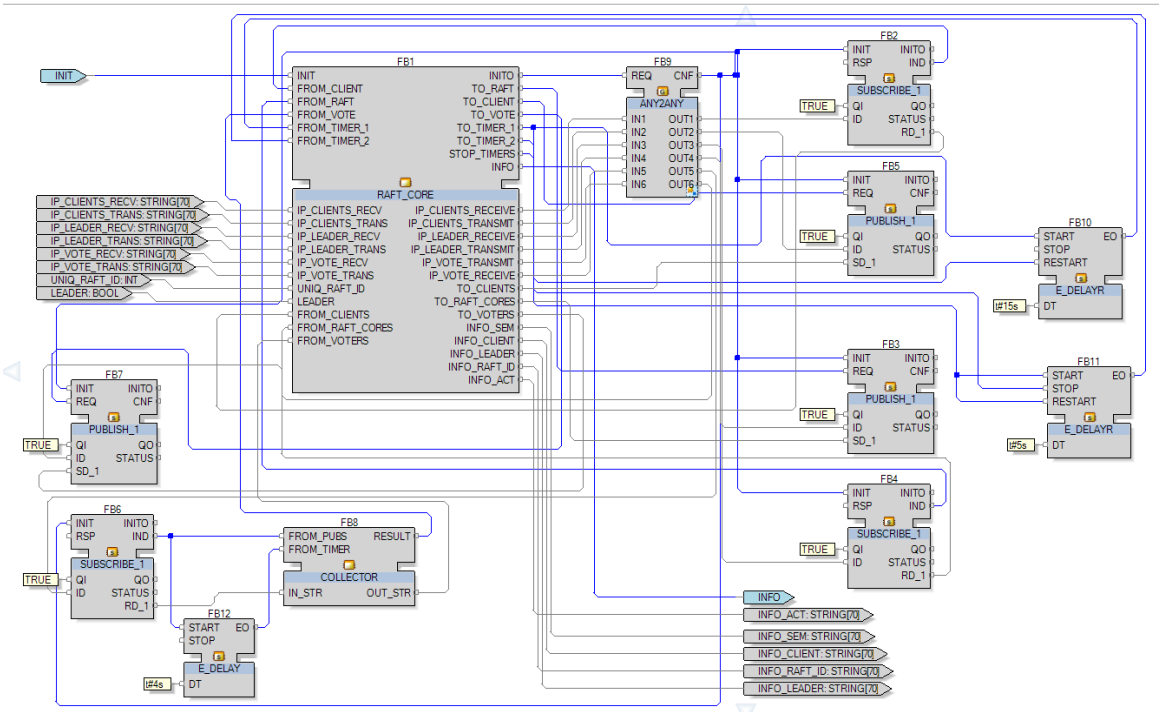


Рис. 2. Структура составного ФБ RAFT

При рассмотрении ситуации отказа компонентов системы также более привлекательным вариантом остается протокол Raft. В него изначально заложен механизм голосования при отказе модуля, работающего в режиме лидера. В результате голосования один из модулей, работающих в режиме подчиненного, становится новым лидером. Протокол Paxos тоже предполагает смену ролей при отказе компонента, например, с ролью Proposer, однако, четко не определяет, модуль с какой ролью должен занять его место.

Таким образом, распределенный семафор, реализованный на основе протокола Raft, обладает следующими преимуществами: 1) простота алгоритма и его реализации; 2) возможность масштабирования системы с минимальными трудозатратами; 3) более высокая производительность.

Предложенная в данной работе функционально-блочная реализация протоколов Paxos и Raft позволяет не только строить на их основе распределенные семафоры, но и обеспечивает корректную работу с распределенными данными и, таким образом, показывает путь реализации распределенных баз данных в системах управления на основе ФБ.

Следует также отметить, что были построены сетевые модели распределенных семафоров на основе Paxos и Raft с использованием цветных временных сетей Петри, а также проведена серия имитационных экспериментов в системе CPN Tools. Направлением дальнейших исследований является функционально-блочная реализация других механизмов синхронизации и взаимодействий, применимых в распределенных управляющих системах.

Библиографический список

1. Vyatkin V. IEC 61499 Function Blocks for Embedded and Distributed Control Systems Design, 3rd edn. / V. Vyatkin. – Pittsburgh : Instrumentation Society of America, 2016. – 261 p.

2. Грегори, Р. Э. Основы многопоточного, параллельного и распределенного программирования : пер. с англ. / Р. Эндрюс Грегори. – М. : Вильямс, 2003. – 505с.

3. Dubinin V. N. Implementation of some kinds of interactions and communications in IEC 61499 architecture / V. N. Dubinin // Новые информационные технологии и системы (НИТиС'2004) : тр. VI Междунар. науч.-техн. конф. (Пенза, Россия, 17–19 июня). – Пенза, 2004. – Ч. 2. – С. 84–92.

4. Lamport, L. Fast Paxos / L. Lamport // Distributed Computing. – 2006. – № 19 (2). – P. 103.

5. Planning for Change in a Formal Verification of the Raft Consensus Protocol / D. Woos, J. R. Wilcox, S. Anton, Z. Tatlock, M. D. Ernst, T. Anderson. – University of Washington, 2016. – 12 с.

6. Yethadka, S. A Distributed Lock Manager Using Paxos: Design and Implementation of Warlock, a Consensus Based Lock Manager / S. Yethadka. – Master thesis, Uppsala University, Department of Information Technology, 2013. – 82 p.

7. nxtStudio. – URL: http://www.nxtcontrol.com

Д. В. Сафронов, А. В. Дубравин

Пензенский государственный университет, Пенза, Россия

АВТОМАТИЗАЦИЯ ПОИСКА ИЗБЫТОЧНЫХ ФУНКЦИОНАЛЬНЫХ ОБЪЕКТОВ В ИСПОЛНИМЫХ ФАЙЛАХ ДЛЯ ПЛАТФОРМЫ MICROSOFT .NET

Рассмотрены вопросы поиска избыточных функциональных объектов в исполнимых файлах для платформы Microsoft .NET, проводимого в рамках исследования программного обеспечения на наличие недекларированных и опасных возможностей.

Руководящий документ ФСТЭК РФ [1] предписывает проверку на наличие избыточных функциональных объектов всего программного обеспечения, используемого на защищенных рабочих местах при обработке информации, содержащей гостайну, и устранение найденной избыточности. Устранение избыточности с целью повышения защищенности информации актуально также и для информационных систем предприятий, не связанных с гостайной, но в которых обрабатывается информация, составляющая коммерческую тайну, и конфиденциальная информация.

Таким образом, актуальной является автоматизация поиска неиспользуемых функциональных объектов в исполнимых файлах для платформы Microsoft .NET.

Под неиспользуемыми функциональными объектами будем понимать методы управляемых классов, вызовов которых (прямых, виртуальных, косвенных, посредством отражения) не содержится в некоторой совокупности исполнимых файлов, которую назовем исследуемым ПО. К неиспользуемым функциональным объектам отнесем также управляемые классы, от которых не наследуется ни одного другого класса, которые не инстанциируются в процессе выполнения исследуемого ПО, и у которых не вызывается ни одного статического метода.

Неиспользуемые функциональные объекты (ФО) могут образовываться в программном коде следующими путями:

– добавление классов и методов для полноты программного интерфейса;

– классы и методы, используемые в специальных сборках (например, отладочных);

– классы и методы, которые использовались ранее.

При компиляции в машинный код во многих случаях избыточный код может быть удален компилятором на стадии оптимизации. При компиляции под управляемые платформу сохраняются все функциональные объекты. При этом сохраняется также информация о типах и сигнатурах этих объектов – метаданные. Это позволяет в процессе выполнения разобрать состав исполнимого файла, выделить определенные типы и методы (например, реализующие заданный интерфейс, имеющие заданное название либо сигнатуру) и вызвать метод либо создать объект. Такие обращения могут осуществляться не только из самого исполнимого файла, но и из других файлов, в том числе, создаваемых позднее. В сре-

де Microsoft .NET механизм, позволяющий разбирать состав исполнимого файла во время выполнения и осуществлять вызовы, называется отражением (Reflection).

Обращения могут производится также из исполнимых файлов, не входящих в состав программного продукта. Так, например, присутствующим в исполнимом файле отладочным кодом могут воспользоваться вредоносные программы (malware).

На данный момент имеется ряд программных продуктов, предназначенных для восстановления и анализа структуры и алгоритмов исполнимых файлов для платформы Microsoft .NET (сборок).

- Дизассемблер ILDasm, разработанный Microsoft, позволяет изучать сборки в интерактивном режиме, представляя структуру файла и алгоритмы отдельных методов, восстановленные до уровня промежуточного языка (MSIL). Навигация возможна только по структуре файла, возможности анализа не предусмотрены. Имеется возможность сохранения всего дизассемблированного кода сборки. Одновременная работа возможна только с одним файлом.
- Декомпиляторы RedGate.NET Reflector и JetBrains dotPeek восстанавливают алгоритмы методов до записи на языке высокого уровня, поддерживают одновременную работу с несколькими файлами. Содержат средства навигации по коду на основе связей между сущностями, интерактивные средства анализа использования функциональных и информационных объектов (переменных, аргументов и полей данных). Декомпиляторы позволяют сохранить восстановленный код в виде проекта, пригодного – как правило, после некоторых доработок – для повторной компиляции.

Имеющиеся средства ориентированы в первую очередь на ручной анализ. Только .NET Reflector позволяет расширять свою функциональность посредством дополнительных модулей – плагинов.

Таким образом, программные средства, позволяющие автоматизировано анализировать состав сборок в соответствии с заданными критериями, отсутствуют.

Функциональный объект-метод класса является используемым, если в коде присутствуют обращения к этому методу одного или нескольких следующих видов:

- прямой вызов статического либо экземплярного метода, в том числе виртуального;
- вызов виртуального метода базового класса (интерфейса), если в коде создается объект рассматриваемого класса и этот объект может появляться в месте вызова;
- получение делегата для метод.

Функциональный объект-класс является используемым, если в коде присутствует:

- создание объекта этого класса;
- вызов статического метода этого класса;
- некоторый метод, принимающий либо возвращающий объект этого класса;
- некоторый тип либо метод, обобщенным параметром (generic) которого является данный класс;
- унаследованный от него другой класс.

Функциональные объекты, не подходящие ни под один из этих критериев, являются неиспользуемыми. Также неиспользуемыми считаются функциональные объекты, используемые только в таких функциональных объектах, которые в свою очередь являются неиспользуемыми. Таким образом, процесс выявления неиспользуемых – избыточных – функциональных объектов является итеративным.

Специальным видом использования является создание объектов и вызов методов посредством механизма отражения (Reflection). При этом, если используемые функциональные объекты идентифицируются именами, факт такого использования может быть детектирован автоматически. Если же функциональные объекты выбираются из некоторого множества в соответствии с определенными критериями, связь их с объявлением в коде не может быть установлена автоматически. В этом случае для обозначения неизбыточности функционального объекта требуется использование дополнительных маркеров. В сборках

для платформы .NET в качестве таких маркеров могут выступать настраиваемые атрибуты (Custom Attributes) специального типа.

В соответствии с вышеизложенным, разработано программное средство автоматизированного анализа исполнимых файлов для платформы Microsoft .NET, которое применяется при проведении тематических исследований.

В качестве входных данных используется группа исполнимых файлов исследуемого ПО, а результатом является перечень избыточных функциональных объектов.

ПО учитывает возможность использования функционального объекта только в модулях, внешних по отношению к тому, в котором этот объект определяется. Такая ситуация характерна, в частности, для библиотек программного кода.

На основании полученного перечня разработчик может удалить избыточные сущности либо провести иную реструктуризацию кода.

В дальнейшем планируется добавление модуля анализа зависимостей, что позволит упростить работу аналитика и существенно сократить объем ручного анализа.

Библиографический список

1. Руководящий документ. Защита от несанкционированного доступа к информации. Программное обеспечение средств защиты информации. Классификация по уровню контроля отсутствия недекларированных возможностей // ФСТЭК России. – 1999. – Ч. 1. – URL: http://fstec.ru/tekhnicheskaya-zashchita-informatsii/dokumenty/114-spetsialnye-normativnye-dokumenty/382-rukovodyashchij-dokument-prikaz-predsedatelya-gostekhkomissii-rossii-ot-4-iyunya-1999-g-n-114

Н. П. Вашкевич, П. Ю. Мясин, С. Ю. Поздняков

Пензенский государственный университет, Пенза, Россия

СТРУКТУРА МНОГОПРОЦЕССОРНОЙ СИСТЕМЫ С АППАРАТНЫМ ПЛАНИРОВЩИКОМ

Рассмотрим многопроцессорную систему, в которой используется планировщик с разделением времени, диспетчер задач на основе глобальной очереди.

Рассмотрим многопроцессорную систему, в которой используется планировщик с разделением времени [2], который использует глобальную очередь задач типа FIFO, разделяемую всеми процессорами (рис. 1). Вновь поступившая задача (процесс) помещается планировщиком в конец очереди, а задачи, находящиеся в начале очереди, являются кандидатами на выполнение. Когда один из процессоров освобождается от текущей работы, он обращается к диспетчеру, который выбирает из «головы» очереди готовую к выполнению задачу и работает с ней до ее завершения или до момента блокирования, например, вследствие необходимости выполнения операции ввода-вывода.

Алгоритм планирования с разделением времени и глобальной очередью задач представляет наиболее простой и одновременно эффективный способ планирования, поскольку обладает рядом достоинств [1]:

– загрузка распределяется равномерно между процессорами, обеспечивая отсутствие простоев процессоров при наличии готовых к выполнению задач;

– простота функционирования, заключающаяся в том, что, когда процессор освобождается, он вызывает функцию назначения задач (диспетчер) из операционной системы.

Эти достоинства обеспечивают преимущественное применение такого способа планирования в многопроцессорных системах.

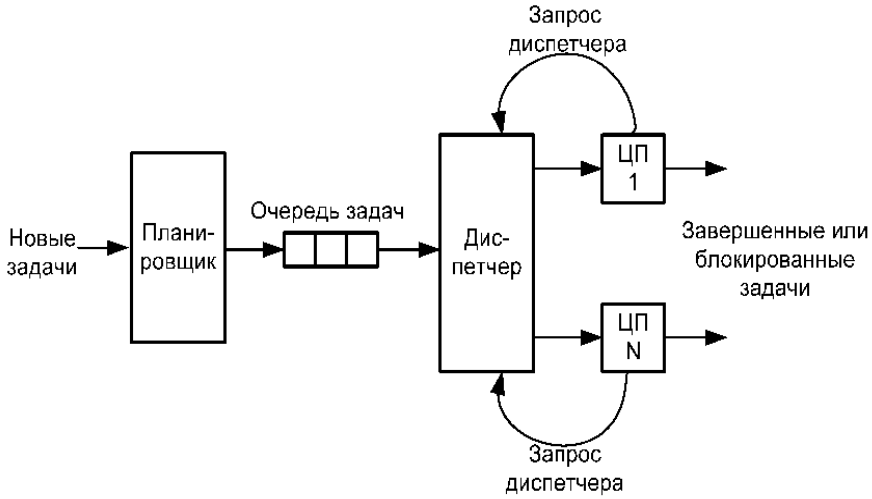


Рис. 1. Модель многопроцессорной системы, реализующей планировщика и диспетчера с глобальной очередью задач

Традиционно планировщик и диспетчер задач выполняются программно и реализуются в SMP-системах методом вызова этих функций из общей памяти, в которой резидентно хранится программа операционной системы. Практическая программная реализация предполагает два варианта:

– в пространстве пользователя;

– в пространстве ядра.

Программная реализация в пространстве пользователя хотя и является быстрой, однако вызывает затруднения при программировании, поскольку для выполнения сопряженной с диспетчеризацией процедуры синхронизации требуется три семафора: один – счетчик – для подсчета числа мест, занятых готовыми для выполнения процессами; другой – счетчик – для подсчета числа процессоров, занятых обслуживанием; и третий – мьютекс – для реализации взаимного исключения, предотвращающего одновременный доступ нескольких свободных процессоров к единственной очереди, которая в данном случае играет роль общего ресурса. Эти обстоятельства могут привести к трудно выявляемым ошибкам непредсказуемым последствиям при работе операционной системы, особенно реального времени.

Очевидным выходом из этого положения является реализация процедуры планирования/диспетчеризации в пространстве ядра [2, 3], однако возрастающие при этом временные затраты на выполнение системных вызовов резко уменьшают производительность многопроцессорной системы.

Одним из путей повышения производительности многопроцессорной операционной системы является аппаратная реализация функций планирования/диспетчеризации [4–6], которая не только снимает проблему временных потерь, но и позволяет увеличить надежность операционной системы, что особенно важно для систем реального времени.

Предлагаемый подход заключается в том, что планировщик и диспетчер задач выполняются в виде независимого аппаратного устройства (специализированного процессора) в составе многопроцессорной системы (рис. 2).

Планировщик создает очередь, состоящую из идентификаторов задач, в соответствии с заданным правилом (FIFO, LIFO, приоритетная очередь) и осуществляет выборку задачи из очереди. Идентификатор задачи однозначно указывает на адрес задачи в памяти вычислительной системы. Традиционно очередь процессов хранится в виде блоков управления процессами (БУП) или *Process Control Block (PCB)* в общей памяти системы по виртуальному адресу либо в специализированной памяти устройства управления процессами (задачами).

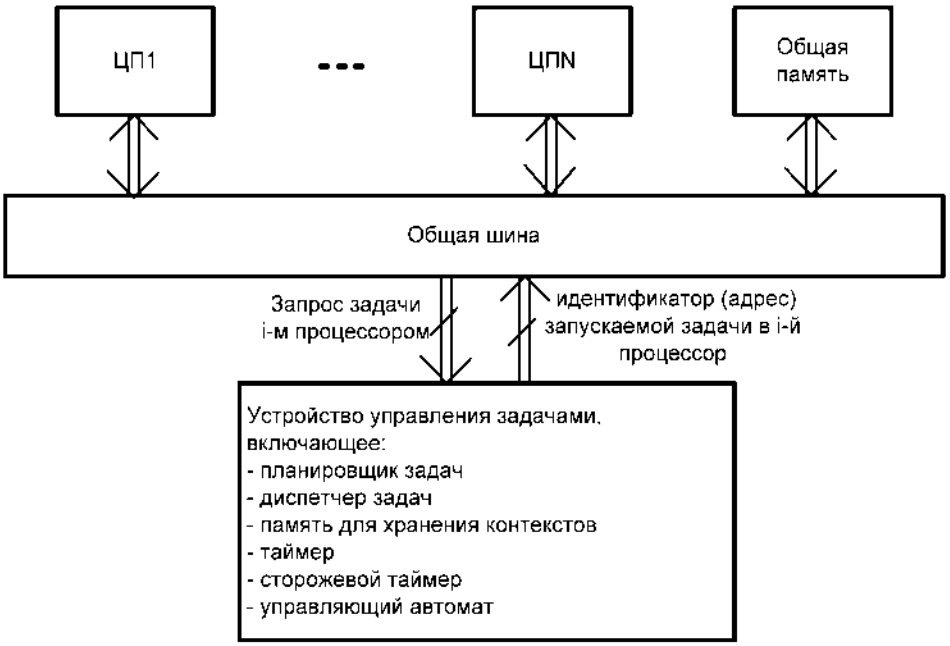


Рис. 2. Схема включения устройства управления задачами в многопроцессорную систему

Диспетчер осуществляет переключение задач (процессов) при прерываниях программ, например, от таймера. Результатом обработки прерывания является сохранение контекста текущей задачи и восстановление контекста задачи, выбираемой из очереди в соответствии с заданным ее типом (FIFO, LIFO, приоритетная). Традиционно контекст хранится в оперативной памяти в виде сегмента состояния процесса (ССП или TSS) и включается в состав БУП.

Таймер предназначен для формирования отметок времени, т.е. квантов процессорного времени, выделяемых каждой задаче из очереди готовых процессов, по которым производится переключение процессов.

Сторожевой таймер отслеживает задачи, которые занимают процессоры на неопределенное время, например, вследствие зацикливания (взаимной блокировки процессов или клинча).

В разработанном аппаратном устройстве управления процессами очередь, создаваемая планировщиком, представляет собой набор из 256 регистров, каждый из которых хранит идентификатор процесса (потока), приоритет процесса, счетчик времени и другую информацию (аналог БУП), необходимую для планирования и диспетчеризации. Каждая процесс имеет собственный идентификатор, который однозначно указывает на адрес задачи в памяти вычислительной системы. Такой подход позволяет значительно ускорить переключение процессов по окончании выделенного кванта времени.

Библиографический список

1. Дейтел, Х. М. Операционные системы. Основы и принципы / Х. М. Дейтел, П. Дж. Дейтел, Д. Р. Чофнес. – М. : Бином-Пресс, 2011. – Ч. 1. – 1024 с.

2. Таненбаум, Э. Современные операционные системы / Э. Таненбаум. – 3-е изд. – СПб. : Питер, 2010. – 1120 с.

3. Эндрюс, Г. Р. Основы многопоточного и распределенного программирования : пер. с англ. / Г. Р. Эндрюс. – М. : Вильямс, 2003. – 512 с.

4. Kuacharoen, P. A configurable hardware scheduler for Real-Time systems / P. Kuacharoen, M. A. Shalan, V. J. Mooney III // In Proceedings of the International Conference on Engineering of Reconfigurable Systems and Algorithms. – Las Vegas, 2003. – P. 96–101.

5. Kohout, P. Hardware support for realtime operating systems / P. Kohout, B. Ganesh, B. Jacob // In Proceedings of the 1st IEEE/ACM/IFIP international conference on Hardware/software codesign and system synthesis. – Newport Beach, CA, USA : ACM, 2003. – P. 45–51.

6. Бикташев, Р. А. Аппаратная поддержка диспетчеризации задач в многопроцессорных системах / Р. А. Бикташев, А. И. Меркурьев // Современные методы и средства обработки пространственно-временных сигналов : сб. ст. VIII Всерос. науч.-техн. конф. – Пенза : Приволжский Дом знаний, 2010. – С. 112.

7. Вашкевич, Н. П. Недетерминированные автоматы и их использование для реализации систем параллельной обработки информации : моногр. / Н. П. Вашкевич, Р. А. Бикташев. – Пенза : Изд-во ПГУ, 2016. – 394 с.

А. В. Калачев

Пензенский государственный университет, Пенза, Россия

РАЗРАБОТКА БАЗОВОГО ФУНКЦИОНАЛЬНОГО БЛОКА УПРАВЛЕНИЯ ДЛЯ СИСТЕМЫ ПЕРЕМЕЩЕНИЯ БАГАЖА В СИСТЕМЕ NXTCONTROL

Разработан базовый блок в соответствии с международным стандартом IEC 61499 для системы управления перемещением багажа с двумя выталкивателями и датчиками состояний, описана диаграмма управления выполнением для разработанного блока, описан его интерфейс взаимодействия с другими компонентами системы, разработаны алгоритмы управления выходными сигналами блока в среде автоматизированного проектирование nxtControl.

В современном мире множество рутинных операций заменяется автоматизированными системами с частичным участием человека или полностью управляемыми компьютерами и роботами.

Оборудование с каждым днем становится быстрее, надежнее, наделяется интеллектуальными функциями управления, благодаря которым система самостоятельно способна оценивать окружающую обстановку и принимать соответствующие решения о взаимодействии с другими объектами и собственной реконфигурации.

Важной особенностью модернизации и замены устаревшего оборудования на более новое, является его совместимость с предыдущими поколениями. На сегодняшний день не все отрасли выработали и перешли на единые стандарты промышленного оборудования обратной совместимости.

В результате модернизации оборудования снижаются издержки производства, затраты на энергетические и экономические ресурсы, что делает целые отрасли более эффективными чем это было ранее. Одной из сфер применения функциональных блоков международного стандарта IEC 61499 являются разработка систем управления при транспортировке и перемещении различных объектов на производстве [1].

К сожалению, не все существующие системы придерживаются международных стандартов, которые позволяют эффективно заменять устаревшее оборудование на более современное т.к. они не используют механизм обратной совместимости. Что приводит к огромным финансовым и трудовым потерям при реконфигурации и замене целых комплексов систем вместо того, чтобы заменить нескольких устаревших модулей на более новые.

В связи с этим актуальным считается разработка автоматизированной системы управления перемещением багажа с использованием международного стандарта IEC 61499 [2], которая в дальнейшем может использоваться в транспортных компаниях, аэропортах, железнодорожных станциях и других областях.

Целью научного исследования является разработка базового функционального блока в соответствии с международным стандартом IEC 61499 для системы управления перемещением багажа с двумя выталкивателями и датчиками состояний, описание ECC диа-

граммы управления выполнением (execution control charts) для базового блока управления, описание его интерфейса взаимодействия с другими компонентами системы, реализация алгоритмов управления выходными сигналами блока в среде автоматизированного проектирование nxtControl.

Полная система управления заданной системой не может быть описана в данной работе, поэтому на примере создания базового блока управления движением двух выталкивателей и функционирования набора датчиков, отвечающих за определение состояний оборудования будет рассмотрен процесс проектирования и реализации других базовых компонентов интеллектуальной системы управления.

Создание базового функционального блока начинается с того, что в созданный проект системы управления на один из ресурсов, например, “RES1”, устройства “Panel_DEV2” добавляется стандартный базовый блок стандарта IEC 61499. Его внешний вид в среде разработки nxtControl представлен на рис. 1 [3].

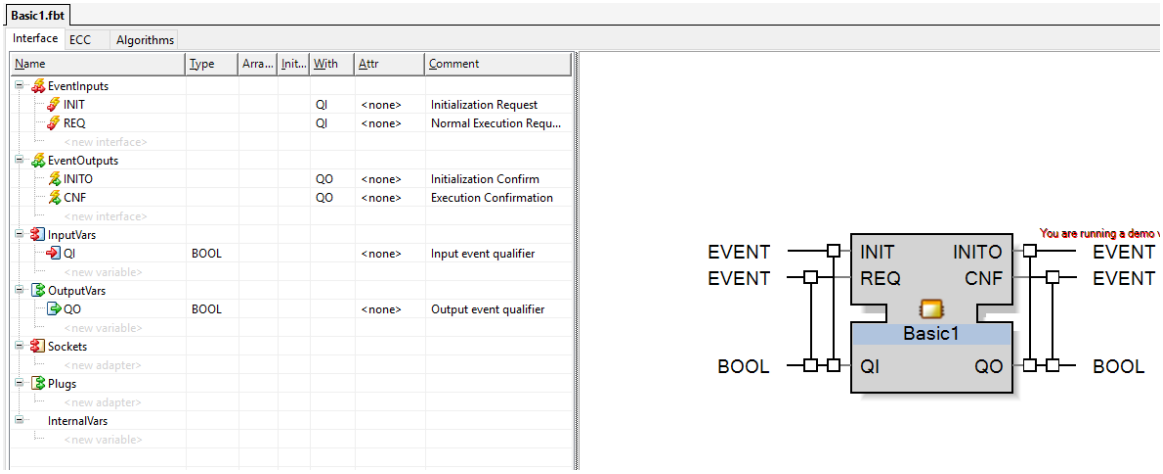


Рис. 1. Внешний вид базового блока и его интерфейс по международному стандарту IEC 61499 в среде nxtControl

Интерфейс стандартного базового блока включает в себя: событийные входы “INIT” и “REQ”, связанные с входной переменной “QI”; событийные выходы “INITO” и “CNF”, связанные с выходной переменной “QO”; входную переменную “QI”; выходную переменную “QO”.

Также у базового блока имеется простая ECC диаграмма управления выполнением. Данная диаграмма позволяет подавать выходные управляющие сигналы в соответствии с логикой работы блока и зависит от входных событийных сигналов, и значений переменных. По умолчанию, в состав ECC диаграммы входят следующие состояния:

- “START” – с него начинается работа функционального блока;
- “INIT” – инициализация параметров блока, при переходе в которое выполняется действие (action) алгоритма с названием “INIT” и подается управляющее воздействие на событийный выход “INITO”;
- “REQ” – нормальное выполнение функций блока, при переходе в которое выполняется действие (action) алгоритма с названием “REQ” и подается управляющее воздействие на событийный выход “CNF”.

А также ECC диаграмма содержит переходы из описанных состояний:

- переход “START” – “INIT” по условию “INIT”;
- переход “INIT” – “START” по условию “1” т.е. всегда выполняется переход;
- переход “START” – “REQ” по условию “REQ”;
- переход “REQ” – “START” по условию “1”.

Созданные по умолчанию алгоритмы “REQ” и “INIT” не содержат в себе специальной логики функционирования блока.

После создания заготовки базового блок управления был переименован в “ControlMove” и выполнена его реконфигурация в соответствии его логикой функционирования [4]. Обновленный интерфейс представлен на рис. 2.

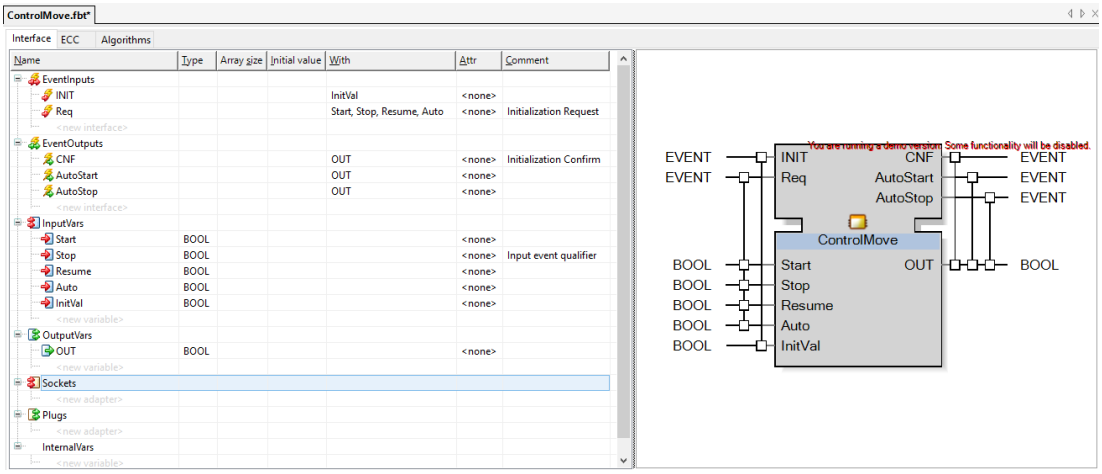


Рис. 2. Внешний вид интерфейса базового блока управления “ControlMove”

Интерфейс базового блока управления **“**ControlMove” включает в себя: событийный вход “INIT”, связанный с входной переменной “InitVal”; событийный вход “Req”, связанный с входными переменными “Start”, “Stop”, “Resume”, “Auto”; событийные выходы “CNF”, “AutoStart” и “AutoStop”, связанные с выходной переменной “OUT”; входную переменную “Start”; входную переменную “Stop”; входную переменную “Resume”; входную переменную “Auto”; выходную переменную “OUT”.

Для базового блока управления двумя выталкивателями и датчиками состояния была разработана ECC диаграмма. Данная диаграмма позволяет осуществлять управляющие воздействия: поднятие багажа, передача его дальнейшему оборудованию, режим блокировки, работа блока управления в ручном и автоматическом режиме в зависимости от входных событийных сигналов и значений переменных. В состав данной ECC диаграммы входят следующие состояния:

- “START” – с него начинается работа функционального блока;
- “INIT” – при переходе в данное состояние выполняется действие (action) алгоритма с названием “INIT” и подается управляющее воздействие на событийный выход “CNF”;
- “Begin” – при переходе в данное состояние выполняется действие (action) алгоритма с названием “StartMove” и подается управляющее воздействие на событийный выход “CNF”;
- “Stop” – при переходе в данное состояние выполняется действие (action) алгоритма с названием “StopMove” и подается управляющее воздействие на событийный выход “CNF”;
- “Resume” – при переходе в данное состояние выполняется действие (action) алгоритма с названием “StartMove” и подается управляющее воздействие на событийный выход “CNF”;
- “AutoOn” – при переходе в данное состояние выполняется действие (action) алгоритма с названием “StartMove” и подается управляющее воздействие на событийный выход “AutoStart”;
- “AutoOff” – при переходе в данное состояние выполняется действие (action) алгоритма с названием “StopMove” и подается управляющее воздействие на событийный выход “AutoStop”.

А также ECC диаграмма содержит переходы из описанных состояний: переход “START” – “INIT” по условию “INIT”; переход “INIT” – “START” по условию “1; переход

“START” – “Begin” по условию “Start”; переход “Begin” – “START” по условию “1”; переход “START” – “Stop” по условию “Stop”; переход “Stop” – “Resume” по условию “Resume”; переход “Resume” – “Begin” по условию “Auto=FALSE”; переход “Resume” – “AutoOn” по условию “Auto=TRUE”; переход “Start” – “AutoOn” по условию “Auto=TRUE”; переход “AutoOn” – “Stop” по условию “Stop”; переход “AutoOn” – “AutoOff” по условию “Auto=FALSE”; переход “AutoOff” – “Start” по условию “1”.

ECC диаграмма и ее алгоритмы для функционирования логики управления выходными сигналами базового блока представлены на рис. 3.

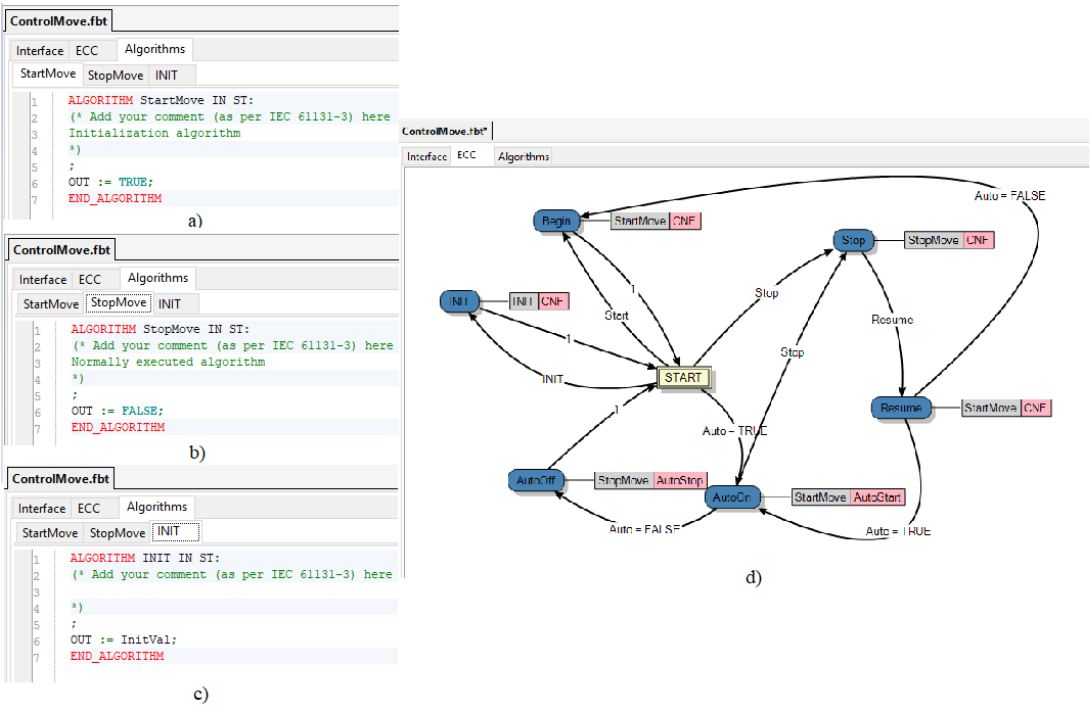


Рис. 3. Внешний вид алгоритмов управления выходными состояниями функционального блока управления “ControlMove”: *a* – StartMove; *b* – StopMove; *c* – INIT; *d* – ECC диаграммы базового блока управления “ControlMove”

Таким образом, была продемонстрирована разработка базового блока в соответствии с международным стандартом IEC 61499 для системы управления перемещением багажа с двумя выталкивателями и датчиками состояний, разработана ECC диаграмма управления для разработанного блока, разработан интерфейс взаимодействия с другими компонентами системы, реализованы алгоритмы управления выходными сигналами блока в среде автоматизированного проектирование nxtControl.

В результате получен базовый блок управления, который является универсальным и в дальнейшем будет встроен в различные системы управления.

Библиографический список

1. Vyatkin, V. IEC 61499 Function Blocks for Embedded and Distributed Control Systems Design / V. Vyatkin // Instrumentation Society of America (ISA) and o^{3}neida. – Auckland University, New Zealand, 2011. – 297 p.

2. Dubinin, V. Formal modeling and verification of IEC 61499 function blocks on the basis of transition systems / V. Dubinin, V. Vyatkin, A. Shalyto // International Siberian Conference on Control and Communications (SIBCON'2016). – M., 2016. – P. 1–4.

3. Интернет-ресурс среды разработки NxtControl. Базовые функции среды разработки. – URL: http://www.nxtcontrol.com/fileadmin/main/download/tutorial_basic_concept_ demo_en.pdf

4. Интернет-ресурс EasyVeep. Система поднятия багажа. – URL: www.easyveep.com

Д. А. Уваров

Пензенский государственный университет, Пенза, Россия

ВСТРАИВАНИЕ БАЗ ЗНАНИЙ В SOA

Рассматриваются процесс управления и встраивания баз знаний, а также реализация сервисно-ориентированной архитектуры с помощью данного процесса.

1. Введение

В настоящем документе предлагается основа для внедрения баз знаний в сервис-ориентированной архитектуре. Есть архитектурная основа для управления бизнес-процессами в сервис-ориентированных системах и набор бизнес-процессов, которые интегрируют управление знаниями с SOA (сервис-ориентированной архитектурой). Важной новизной структуры является то, что она позволяет автоматизировать эволюцию уровня знаний в базе знаний и интегрировать эволюцию базы знаний концептуального уровня с общей сервис-ориентированной системой. В рамках этой рамки решается ряд других вопросов в области управления знаниями, таких как интеграция в повседневную работу и сбор информации из разных ресурсов. Фокусировка происходит на онтологических системах управления знаниями, которые стали одной из важнейших технологий внедрения систем управления знаниями [1].

2. Автоматизация бизнес-процессов в SOA

В последнее десятилетие сервис-ориентированная архитектура стала наиболее признанным подходом к автоматизации бизнес-процессов. В нем основное внимание уделяется тому, как информационные технологии должны поддерживать бизнес-процессы организации с ее основной мотивацией для увеличения возможностей организации в краткосрочной перспективе решать новые бизнес-требования за счет повторного использования существующей бизнес-логики и моделей данных, что приведет к минимальным затратам, ресурсам и накладным расходам времени, минимизируя риски, особенно по сравнению с переписыванием целых прикладных систем. Для этого он добавляет новый уровень обслуживания между уровнем бизнес-процессов и прикладным уровнем предприятия. Сервисный уровень состоит из сервисов, единиц логики, которые должны основываться на сервис-ориентированных принципах, таких как свободное соединение, контракт на обслуживание, автономность, абстракция, многоразовость, возможность компоновки и открытость. Одним из преимуществ уровня сервиса является то, что возможность компоновки позволяет дополнительно структурировать сервисы на разных уровнях. Рекомендуется, чтобы более низкие уровни обслуживания состояли из сервисов приложений и что верхний уровень состоит из сервисов процесса. Сервисы процессов выполняют согласование сервисов нижнего уровня, чтобы их операции выполнялись в определенном порядке. Таким образом, происходит автоматизация бизнес-процесса, что является важной частью достижения лучшей гибкости бизнес-системы.

В настоящее время сервис-ориентированная архитектура основана на технологии веб-сервисов. Для организации веб-сервисов требуется модель, с помощью которой бизнес-процессы можно описать с точки зрения подключения веб-сервисов. Эта модель предоставляется BPEL (язык исполнения бизнес-процессов), язык на основе XML, предназначенный главным образом для поддержки автоматизированных бизнес-процессов на основе веб-сервисов.

3. Управление знаниями и онтологии

Важная часть подходов к управлению знаниями основана на онтологиях. Онтология – это общее понимание некоторой области интереса. Это как средство обеспечения обмена

информацией и знаниями путем сбора общего понимания терминов, которые могут использоваться как людьми, так и программами. Существует язык веб-онтологии OWL. Он состоит из отдельных лиц, свойств и классов. Отдельные экземпляры представляют объекты в интересующем домене. Свойства – это двоичные отношения между отдельными лицами. Классы OWL интерпретируются как наборы, содержащие людей. Они определяются с использованием формальных описаний, которые точно определяют требования к членству в классе. Документ OWL может использоваться для представления двух аспектов организации: концептуального уровня (онтологии) и уровня экземпляра.

Решения по управлению знаниями основаны на организационной памяти, которая объединяет неформальные, полуформальные и формальные знания для облегчения доступа, совместного использования и повторного использования членами бизнес-системы для решения их индивидуальных или коллективных задач [2]. Концепция базы организационных знаний (OOKB), относится к хранимой информации из истории организации и поддерживает совместное использование и повторное использование индивидуальных и корпоративных знаний и уроков.

4. Встраивание баз знаний через автоматизацию бизнес-процессов

Встраивание баз знаний способствует захвату, хранению и распространению знаний с использованием информационных технологий. Встраивание баз знаний рассматривается как непрерывный процесс, который создает знания как побочный продукт работы, интегрирует знания в открытый и развивающийся репозиторий и распространяет его среди других в организации, когда он имеет отношение к их работе. Предлагаемая структура основывается на этих наблюдениях. В контексте бизнес-процесса они переходят к следующему: это непрерывный процесс, который создает знания как побочный продукт выполнения бизнес-процесса, интегрирует знания в открытый и развивающийся репозиторий и распространяет его среди других в организации, когда этого требуют процессы.

С распространением SOA и связанных с SOA технологий наблюдаются несколько возможностей:

1. Автоматический перевод данных процесса в OOKB. Вся бизнес-деятельность для данной бизнес-системы может быть описана бизнес-процессами. Таким образом, каждый информационный компонент, релевантный и известный бизнес-системе, обрабатывается по меньшей мере в одном из бизнес-процессов. OOKB входит в информационную систему и что информационная система автоматизирует один или несколько бизнес-процессов в отношении принципов сервис-ориентирования. В контексте технологических процессов данные процесса имеют определенный смысл и поэтому должны соответствовать информации о случаях в OOKB. Следовательно, существует трансляция между данными процесса и уровнем экземпляра OOKB. В SOA-системе данные уровня экземпляра могут быть захвачены посредством выполнения бизнес-процесса, если такие переводы определены.

Одним из важных видений SOA является автоматизация значительной части бизнес-процессов таким образом, чтобы обеспечить гибкое, экономичное и эффективное внедрение изменений. В полностью реализованной SOA-системе выполнение всех бизнес-процессов будет автоматизировано с помощью механизма выполнения бизнес-процессов, организующего различные сервисы. В такой системе каждый экземпляр OOKB будет иметь соответствующий источник в бизнес-процессе и может быть автоматически переведен в OOKB.

Данные являются основой для информации. Они представляют факты, символы, которые описывают некоторый контент, без более широкого контекста. Информация представляет обработанные данные, которые имеют определенное добавленное значение для пользователя. В контексте выполнения бизнес-процесса данные процесса являются основой для входной информации, необходимой для выполнения действий процесса. Чтобы использовать их вместе с информацией в OOKB и дать возможность обосновать их, нужно привести их в правильный контекст. Этот контекст определяется моделью бизнес-процессов бизнес-системы. Кроме того, в разных экземплярах процесса и активности разные данные могут представлять различную информацию в зависимости от субъекта, который выполняет деятельность. Это особенно важно в случае человеческих задач. Контекст

также зависит от субъекта, ответственного за деятельность, например, индивидуума, отдела, бизнес-системы в целом или другого бизнес-объекта.

Данные процесса соответствуют уровню экземпляра OOKB. Чтобы поместить данные процесса в правильный контекст, определяются переводы в OOKB. Целью является автоматическое получение информации о экземплярах OOKB во время выполнения бизнес-процесса. Возможно, потребуется подтверждение и подтверждение. В этом случае перевод является полуавтоматическим.

Структура данных процесса определяется схемами XML. Таким образом, основой для определения перевода является XML-схема данных процесса и уровень онтологии OOKB. Все данные процесса не всегда релевантны для выполнения процесса. Можно ссылаться на данные процесса, которые представляют входные данные в качестве носителей информации процесса. Информационный носитель процесса может быть любой частью данных процесса. Он может быть представлен одним XML-элементом, появлением элемента внутри другого элемента, комбинацией нескольких элементов и т.д.

2. Сбор информации из разных источников с использованием SOA-подхода к интеграции: сервис-ориентированная интеграция использует решения на основе совместимых сервисов. Из-за нейтральной для поставщиков инфраструктура связи, основанная на SOA, основанной на веб-сервисах, потенциал для предприятий заключается в реализации высоко стандартизированных описаний услуг и структур сообщений. Таким образом, сбор информации из разных источников в SOA осуществляется в отношении принципов ориентации на обслуживание и использования одной и той же интеграционной платформы. Например, предполагается, что информационные системы, содержащие информацию, необходимую для действия, выставляют веб-сервис с операцией, которая возвращает требуемую информацию. В этом случае сбор информации может быть реализован с помощью процесса BPEL, вызывающего соответствующие сервисы. Необходимая информация будет получена через выход сервиса.

Подход SOA к интеграции может также использоваться для сбора информации из разных источников, а не для сбора информации из разных источников таким образом, чтобы обеспечить плотную связь с используемой системой. Для этого сам OOKB может быть представлен как веб-сервис. Можно обеспечить большую гибкость, когда исходные информационные системы меняются во времени, а также независимость конкретной используемой системы. Поэтому предлагаемая структура основывается на принципах ориентации на сервисы и состоит из нескольких слабо связанных сервисов, которые соответствуют этим принципам.

3. Автоматические запросы базы знаний OWL. В SOA-системе бизнес-процессы реализуются как процессы BPEL, которые выполняют различные сервисы, необходимые для выполнения процесса. Для этого необходимо определить точные модели бизнес-процессов и данные процесса. В случае деятельности крайне непредсказуемого характера все компоненты данных будут трудно определить заранее. Однако во многих случаях может быть идентифицирована деятельность бизнес-процесса, для которой требуется OOKB вместе с необходимыми компонентами данных. Для этих действий требуемый запрос OOKB может быть указан статически или динамически. Во время выполнения они могут выполняться автоматически.

4. Интеграция в повседневную работу. Формализация бизнес-процессов позволяет идентифицировать значительную часть мероприятий, требующих OOKB, вместе с требуемым информационным компонентом. Это особенно полезно в случае действий, выполняемых человеческими задачами. При определении человеческой задачи часто можно определить, какие компоненты данных понадобятся для ее выполнения. Поэтому запросы OOKB могут быть частью определения человеческой задачи. Перед тем, как экземпляр задачи будет представлен лицу, выполняющему задачу, запрос может быть выполнен. Полученная информация может быть представлена владельцу задачи с заданием. Таким образом, для владельца задачи использование OOKB является прозрачным.

Когда деятельность по эволюции онтологии требует изменений в онтологии, то они должны реализовываться. Эти действия являются частью бизнес-процесса эволюции онто-

логии. Поскольку они требуют человеческой работы, они могут быть реализованы как человеческие задачи. Когда необходимо внести изменения в OOKB, задача человека запускается и назначается. События, которые инициируют эти действия и как они должны выполняться, определяются с помощью выбранной методологии. Развитие и эволюция онтологии управляются XML-схемой.

5. Автоматизация определенных человеческих задач, основанных на знаниях. Если знания о действиях захватываются способом, который может быть автоматически обработан, и если доступны необходимые входы активности, задача может быть выполнена автоматически. Действия, требующие проведения логического процесса, редко автоматизируются, поскольку они требуют человеческих рассуждений и логических способностей. Однако часто рассуждающие действия требуют аргументации, которая носит повторяющийся характер - разные экземпляры деятельности требуют одного и того же или аналогичного процесса рассуждения. В бизнес-системах владельцы человеческих задач также ограничены политикой и правилами, которые они должны соблюдать. Для повторяющихся типов методов рассуждений существуют методы, которые могут быть использованы для захвата этого типа знаний, например, правил принятия решений, бизнес-правил, логических инструкций и т.д. На основании этого требуемый ввод для аргументации может быть извлечен из соответствующего источника, если он доступен. Поэтому, если знания, необходимые для аргументации, фиксируются в OOKB, задача анализа может быть автоматизирована путем выполнения запроса и аргументации с помощью OOKB. Это означает, что при соответствующем планировании, выборе соответствующих методов и применении их к системе, основанной на онтологии, как части общей системы SOA, может быть достигнут более высокий уровень автоматизации бизнес-процессов.

Библиографический список

1. Ontologies for Enterprise Knowledge Management / A. Maedche, B. Motik, L. Stojanovic, R. Studer, R. Volz // IEEE Intelligent Systems. – 2003. – № 18 (2). – P. 26–33.

2. Dieng, R. Methods and tools for corporate knowledge management / R. Dieng, O. Corby, A. Giboin, M. Ribiere // International Journal of Human Computer Studies. – 1999. – № 51 (3). – P. 567–598.

И. А. Беззатеев, В. Н. Дубинин, Д. А. Уваров

Пензенский государственный университет, Пенза, Россия

МОДЕЛИРОВАНИЕ РАСПРЕДЕЛЕННЫХ СЕМАФОРОВ НА ОСНОВЕ ПРОТОКОЛОВ PAXOS И RAFT В СИСТЕМЕ CPN TOOLS

Разработаны сетевые имитационные модели (в виде цветных временных сетей Петри) распределенных семафоров, основанных на алгоритмах Paxos и Raft для решения задач консенсуса в сети ненадежных процессов, проведены имитационные эксперименты в системе CPN Tools, даны сравнительный анализ и рекомендации по их использованию. Делается вывод, что семафор на основе Raft существенно производительнее семафора на основе Paxos.

Одним из наиболее важных примитивов синхронизации и взаимодействий процессов в параллельных системах является семафор. Семафор – это особый тип разделяемой переменной, которая обрабатывается только двумя неделимыми операциями – P и V (за-

крытие и открытие семафора) [1]. На основе семафоров строятся другие, более сложные примитивы и задачи синхронизации. Например, задача «производитель-потребитель» реализуется с использованием трех семафоров. По своей сути семафор относится к методам взаимодействий процессов через общую (разделяемую) память, поэтому существует семантический разрыв между концепцией семафора и концепцией распределенных систем, где взаимодействия осуществляются посредством передачи сообщений.

Существует несколько подходом к реализации распределенных семафоров. Один из подходов использует двухфазный протокол блокировки в распределенных базах данных с дублированием копий [2]. Однако в представленной работе будет использоваться более современный подход на основе использования алгоритмов (протоколов) для решения задач консенсуса в сети ненадежных процессов – Paxos [3] и Raft [4]. Консенсус – процесс получения согласованного результата группой участников, основная проблема – наличие помех в среде передачи данных. Алгоритм Raft разрабатывался с учетом недостатков алгоритма Paxos. При выборе ключевых идей предпочтение отдавалось более простым и практичным решениям.

Целью данной работы является моделирование распределенных семафоров на основе протоколов Paxos и Raft с целью верификации, а также для их сравнительного анализа по производительности. Наиболее подходящим методом исследования является имитационное моделирование. В рамках этого метода с точки зрения формализации наиболее подходящими являются модели на основе сетей Петри, позволяющие описывать параллельные взаимодействующие процессы. В работе [5] был предложен сетевой формализм (СФ), позволяющий строить формальные модели на основе расширенных сетей Петри, разработана система сетевого имитационного моделирования (СИМС), однако отсутствие версии СИМС на современных компьютерах не позволяет использовать СФ на практике. Поэтому для формального описания протоколов Paxos и Raft были выбраны цветные (раскрашенные) временные сети Петри [6], положенные в основу инструментальной системы моделирования CPN Tools [7]. Цветные временные сети Петри и система CPN Tools хорошо зарекомендовали себя для моделирования коммуникационных протоколов [8].

Имитационная модель семафора должна обеспечивать следующий функционал: а) генерацию запросов на захват или освобождение семафора от нескольких клиентов; б) обработку и анализ запросов клиентов в соответствии с протоколом; в) отправку клиенту ответа на его запрос; г) сбор статистических данных при проведении моделирования (время захвата или освобождения семафора, время получения клиентом ответа на запрос, данные пакетов, полученных клиентом). Ввиду того, что сетевые модели Paxos и Raft довольно громоздкие, их описание приводится ниже лишь в ограниченном объеме.

Разработанная модель системы «Распределенный семафор на основе протокола Paxos» включает собой раскрашенную сеть Петри, имитирующую работу протокола Paxos для достижения консенсуса при одновременном доступе клиентов к критическому ресурсу, она имеет иерархическую структуру и содержит в себе следующие подсети: System – сеть самого верхнего уровня, включает в себя все нижеперечисленные подсети; Client, Client 2 – подсети, имитирующие генерацию запросов на открытие/закрытие семафора от соответствующих клиентов и прием ответов; SubLearner – подсеть, которая обрабатывает пакеты, приходящие от клиентов, анализирует их и выносит решение об отправке пакета в подсеть Proposer или отправляет клиенту ответ о невозможности выполнения его запроса; Proposer, Acceptor 1, Acceptor 2, Acceptor 3, Learner – подсети, имитирующие работы соответствующих ролей протокола Paxos. Сеть System, которая представлена на рис. 1, является самой верхней в иерархической структуре и содержит в себе остальные подсети.

Имитационные эксперименты с разработанными сетевыми моделями в системе CPN Tools показали работоспособность и эффективность алгоритмов Paxos и Raft для решения задачи достижения консенсуса в процессе взаимодействия клиентов с критическим ресурсом. В ходе моделирования были получены статистические данные о работе модели. На основе их анализа были выявлены следующие преимущества и недостатки разработанных систем (на основе Paxos и Raft): а) обе сети успешно выполняют обработку запросов на захват или освобождение семафора от нескольких клиентов; б) в обеих сетях преду-

смотрен механизм решения проблемы отказа компонента системы и в ходе тестирования разработанных систем была доказана его работоспособность.

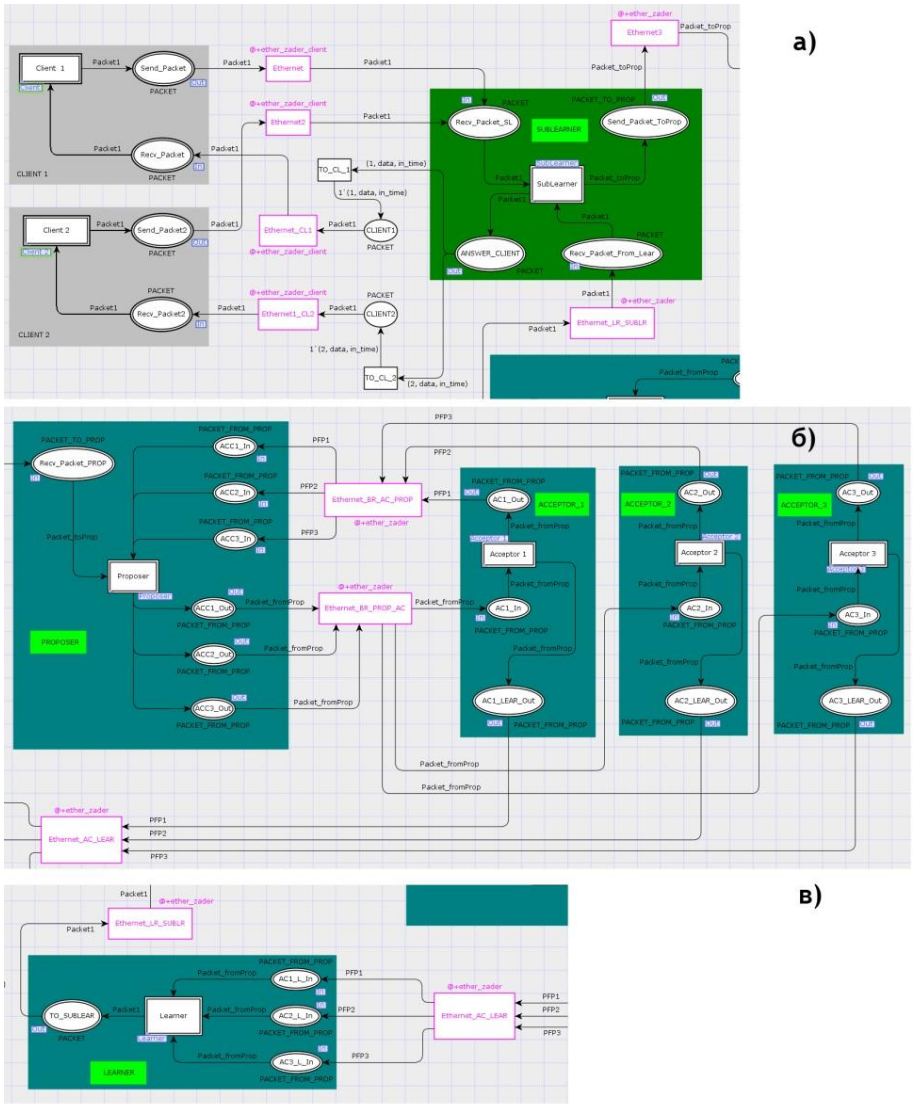


Рис. 1. Сетевая модель верхнего уровня (протокол Paxos)

При проведении имитационного моделирования осуществлялся сбор следующих статистических данных: а) минимальное, среднее, максимальное время выполнения операции захвата или освобождения семафора; б) минимальное, среднее, максимальное время получения клиентом ответа на отправленный запрос; в) количество запросов, пришедших от клиентов, в модули; г) информация о состоянии семафора после успешного выполнения клиентского запроса.

В ходе анализа данных, полученных в процессе имитационного моделирования моделей распределенного семафора на основе протоколов Paxos и Raft, были построены соответствующие таблицы и гистограммы. В приведенной таблице представлены данные, полученные в результате анализа информационных пакетов, генерируемых при успешном выполнении операции захвата или освобождения семафора (подсети SubLearner, Leader_RAFT).

На рис. 2 представлены следующие гистограммы (в порядке следования): «Время получения клиентом ответа на свой невыполненный запрос», «Время получения клиентом ответа на свой выполненный запрос», «Время выполнения операции захвата/освобождения семафора»).

Время выполнения операции захвата/освобождения семафора		
Количество величин в заданном диапазоне	PAXOS	RAFT
3–5 мс (шт.)	0	5836
5–8 мс (шт.)	5418	866
8–14 мс (шт.)	1158	0
более 14 мс (шт.)	97	0
Минимальное время	6,89 мс	3,13 мс
Среднее время	7,846 мс	3,781 мс
Максимальное время	18,12 мс	5,26 мс

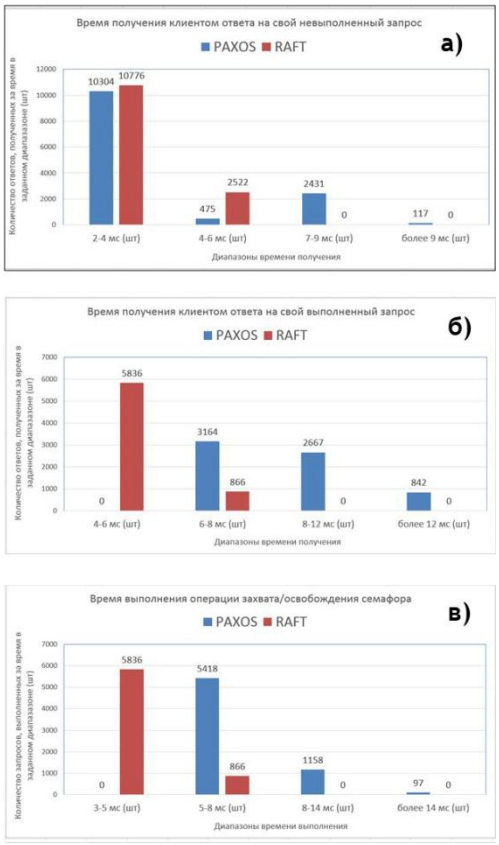


Рис. 2. Результаты имитационного моделирования

Из рис. 2,а видно, что основное количество невыполненных запросов, обрабатываемых сетями Paxos и Raft, обрабатываются за время 2–4 мс. Однако в сети Paxos часть запросов была обработана за время большее чем 7 мс, а в сети Raft данная величина не превышала 6 мс. На рис. 2,б аналогичная тенденция просматривается с количеством выполненных запросов. На рис. 2,в показано, что и сама операция захвата или освобождения семафора, в целом, выполняется быстрее в сети на основе алгоритма Raft.

Исходя из результатов, полученных при имитационном моделировании моделей распределенного семафора на основе протоколов Paxos и Raft в среде CPNTools, можно сделать следующие выводы: а) в сети на основе Raft клиенты получали ответ на свой невыполненный запрос в среднем на 21,1 % быстрее, чем в сети на основе Paxos; б) в сети на основе Raft клиенты получали ответ на свой выполненный запрос в среднем на 46 % быстрее, чем в сети на основе Paxos; в) в сети на основе Raft запрос на захват или освобождение семафора выполнялся в среднем на 51,8 % быстрее, чем в сети на основе Paxos; г) обе сети успешно выполняют обработку запросов на захват или освобождение семафора от нескольких клиентов.

Система «Распределенный семафор», реализованная с использованием алгоритма Raft обладает следующими преимуществами: а) простота алгоритма и его реализации;

б) возможность масштабирования системы с минимальными трудозатратами; в) система справляется с поставленной задачей быстрее. В общем, использование алгоритма Raft при построении системы распределенного семафора является более простым и эффективным способом, чем использование протокола Paxos.

Направлением дальнейших исследований является построение распределенных семафоров на основе других принципов, алгоритмов и протоколов, их моделирование и сравнительный анализ.

Библиографический список

1. Эндрюс, Г. Р. Основы многопоточного, параллельного и распределенного программирования : пер. с англ. / Г. Р. Эндрюс. – М. : Вильямс, 2003. – 505 с.

2. Дубинин, В. Н. Инструментальные средства для проектирования, верификации и реализации распределенных вычислительных систем на основе локальной сети / В. Н. Дубинин // Вычислительная техника в автоматизированных системах контроля и управления : Межвуз. сб. науч. тр. – Пенза : Изд-во ПГТУ, 1996. – Вып. 24. – С. 43–49.

3. Алгоритм Paxos. – URL: https://ru.wikipedia.org/wiki/Алгоритм_Паксос

4. Алгоритм Raft. – URL: https://ru.wikipedia.org/wiki/Алгоритм_Raft

5. Дубинин, В. Н. Сетевые модели распределенных систем обработки, хранения и передачи данных : моногр. / В. Н. Дубинин, С. А. Зинкин. – Пенза : Приволжский Дом знаний, 2013. – 452 с.

6. Jensen, K. Coloured Petri Nets. Modelling and Validation of Concurrent Systems / K. Jensen, L. M. Kristensen. – Springer Publishing Company, 2009. – 384 p.

7. CPN Tools. – URL: http://cpntools.org/

8. Зайцев, Д. А. Моделирование телекоммуникационных систем в CPN Tools / Д. А. Зайцев, Т. Р. Шмелева. – Одесса : ОНАС, 2008. – 68 с.

И. В. Сенокосов

Пензенский государственный университет, Пенза, Россия

ФУНКЦИОНАЛЬНО-БЛОЧНАЯ РЕАЛИЗАЦИЯ МАГАЗИННЫХ АВТОМАТОВ

Предлагается метод реализации магазинных автоматов на основе функциональных блоков международного стандарта МЭК 61499. Предложенный реализационный подход может быть использован в проектировании систем управления выборкой деталей в производственных системах.

В настоящее время в связи с усложнением технологических процессов и требованием к их быстрой перенастройке возникает потребность как в формализации техпроцессов, так и в разработке на основе формальных описаний методов реализации систем управления, поддерживающих данные техпроцессы. Наиболее перспективный подход к построению реконфигурируемых распределенных систем управления предлагает международный стандарт МЭК 61499 [1]. Наибольшее распространение при построении систем управления технологическим оборудованием нашли конечные автоматы и регулярные языки. В работе [2] предложен подход к реализации недетерминированных конечных автоматов на основе функциональных блоков (ФБ) стандарта МЭК 61499. Следует отметить, что конечные автоматы являются достаточно простыми моделями и не могут описать сложные процессы, например, представляемые контекстно-свободными (КС) языками. Для распознавания КС-языков используются магазинные автоматы (МА) [3]. Формальное определение МА можно найти, например, в [4]. В отличие от конечных автоматов, в МА используется магазинная память (стек, магазин).

В данной работе для простоты будут использоваться только детерминированные МА. Пример графического представления МА в человеко-машинном интерфейсе, реализованным в *NxtStudio* [5], приведен на рис. 1. Пометка каждого перехода показывает входной символ, вершину стека, а также чем она будет заменена. Следует заметить, что сам стек в данном представлении не фигурирует. Приведенный на рис. 1 МА распознает КС-язык, заданный следующей формальной грамматикой:

$$S \rightarrow S + T \mid T$$

$$T \rightarrow (S) \mid a.$$

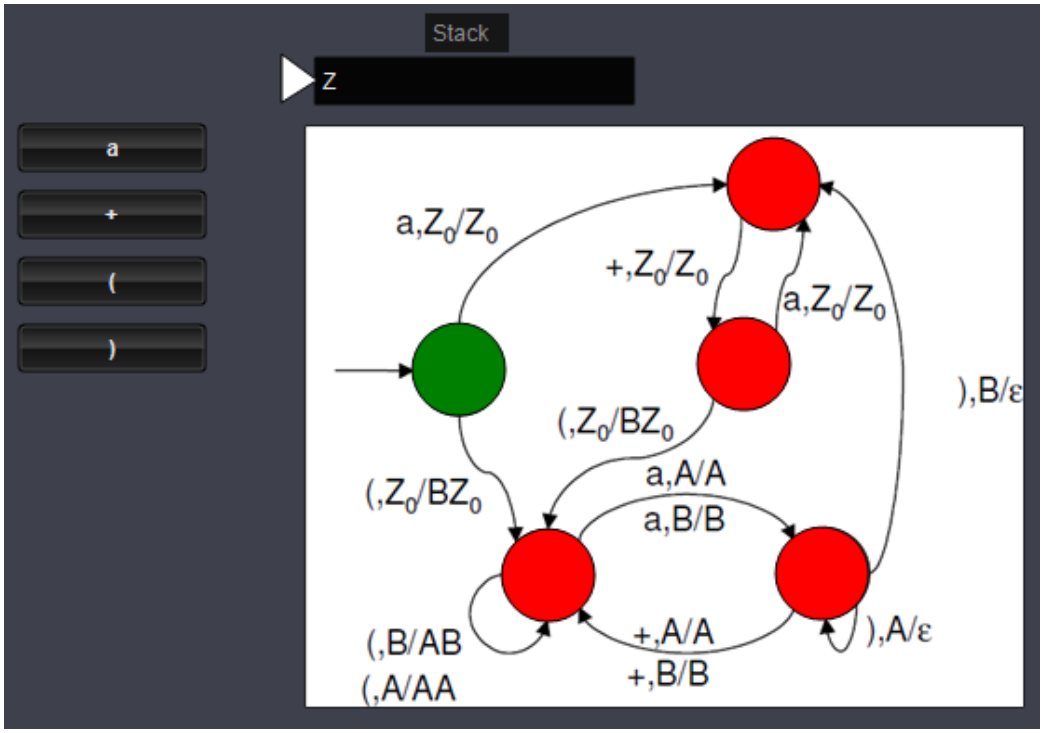


Рис. 1. Графический интерфейс для моделирования работы МА

Данная грамматика определяет язык скобочных арифметических выражений с операцией сложения над операндом *a*.

Рассмотрим подход к реализации магазинных автоматов на основе функциональных блоков (ФБ) международного стандарта МЭК 61499 [1]. Данный подход предполагает реализацию МА, когда переходы представляются в виде ФБ. При данном подходе ФБ, представляющий переход, одного и того же типа и отличается только некоторыми параметрами при инициализации. Можно также представлять состояния в виде ФБ, но каждый ФБ будет отличаться числом входных/выходными сигналов и переменных, и сложность реализации каждого отдельного блока-состояния будет довольно большой.

Для моделирования функционирования МА используется механизм маркеров. Маркер является динамическим объектом, который может быть передан из одного ФБ-перехода в другой ФБ-переход. Если ФБ имеет маркер, то он может принимать входные символы. Упрощенная схема реализации представлена на рис. 2,а. Полная схема реализации примера из рис. 1 не приведена из-за большого объема – в работающей схеме тринадцать ФБ-переходов. Первый блоков типа *StateMA* реализует один из переходов автомата. ФБ *Switch* управляет потоками данных от ФБ-переходов в функциональный блок стека. ФБ *Stack* отвечает за все операции связанные со стеком (магазином). Реализация МА функционирует следующим образом: приходит некоторый входной сигнал, срабатывает один из ФБ-переходов или несколько ФБ, если ФБ принимают один и тот же входной сигнал и в них установлены маркеры. Далее сигналы через ФБ *Switch* направляются в магазин. Если

верхний магазинный символ совпал с магазинным символом перехода, то сработавший ФБ-переход передает свой выходной сигнал на один из выходов $S_0 \ldots S_z$.

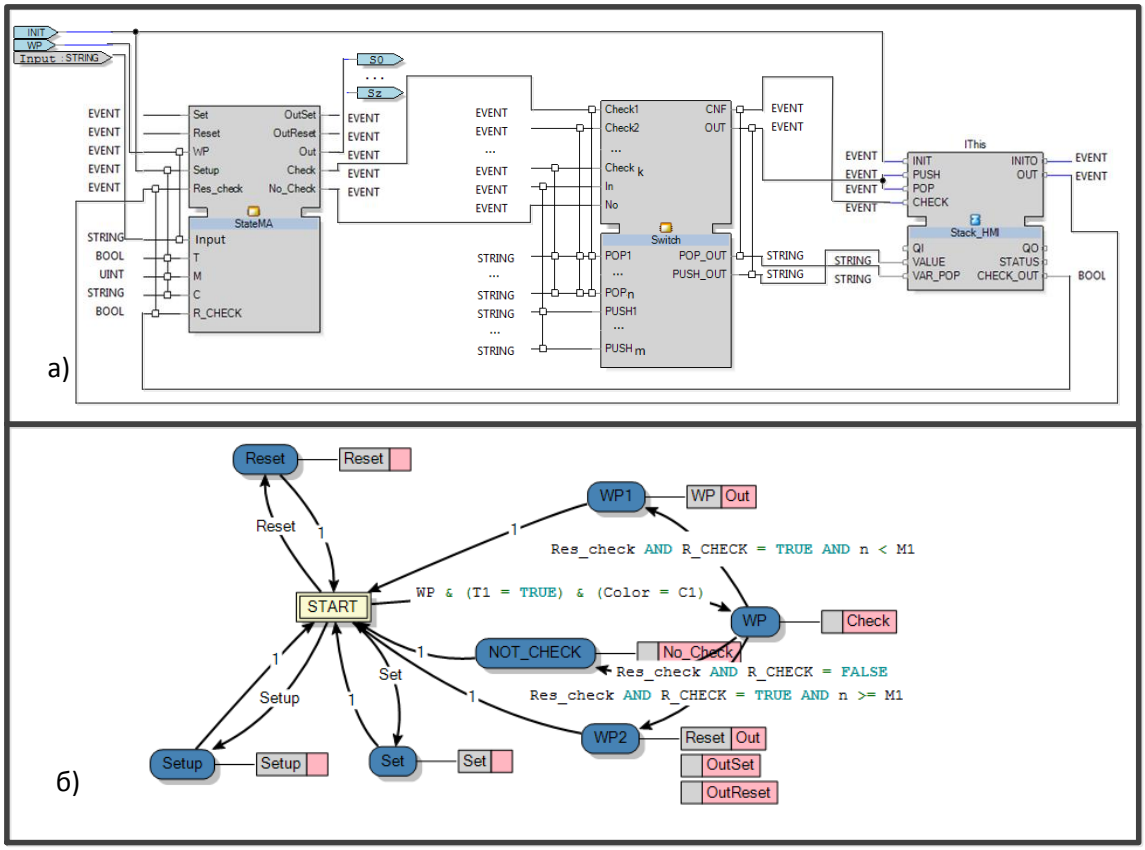


Рис. 2. а – сеть ФБ, реализующая МА; б – диаграмма ECC ФБ StateMA

Рассмотрим реализацию ФБ *StateMA* подробнее. Интерфейс данного базисного ФБ состоит из следующих групп параметров:

Входные сигналы:

Set – сигнал установки маркера в данном ФБ;

Reset – сигнал удаления маркера из данного ФБ;

WP – сигнал поступления входного символа;

Setup – сигнал, установки (сброса) начальных параметров ФБ;

Res_Check – сигнал проверки верхнего символа магазина с магазинным символом ФБ.

Выходные сигналы:

OutSet – сигнал, устанавливающий маркеры в других ФБ;

OutReset – сигнал, сбрасывающий маркеры в других ФБ;

Out – сигнал, означающий, что ФБ сработал;

Check – сигнал проверки верхнего символа магазина;

No_Check – сигнал, который выдается, если верхний символ магазина не совпал с магазинным символом ФБ.

Входные переменные:

Input – входной символ;

C – входной символ, который может принять ФБ;

T – флаг наличия маркера в ФБ;

M – максимальное количество деталей заданного цвета, которые могут быть обработаны данным ФБ перед передачей маркера на другой (другие) ФБ.

R_CHECK – результат проверки верхнего символа магазина.

Диаграмма управления выполнением (иначе, диаграмма ECC) базисного ФБ *State-MA* представлена на рис. 2,б. и состоит из шести состояний:

1) состояние *Reset*, в котором выполняется сброс маркера ФБ;

2) состояние *Setup*, в котором устанавливаются начальные значения внутренних переменных;

3) состояние *Set*, в котором происходит установка маркера ФБ;

4) состояние *WP*, в которое автомат переходит при выполнении следующего условия: *WP & (T1 = TRUE) & (Color = C1)* WP&(T1 = TRUE)&(Color = C1). Переход в данное состояние произойдет, если в реализацию МА поступит входной символ, воспринимаемый этим ФБ, и данный ФБ имеет маркер, при переходе в данное состояние выдается сигнал *Check*;

5) состояние *WP1*, в котором происходит увеличение счетчика количества прохождений по петле. В данное состояние может попасть ФБ, *R_CHECK = TRUE* и *M > 1*. Кроме того, в данном состоянии выдаются выходные сигналы *Out* и *OutReset*;

6) состояние *WP2*, в котором происходит удаление маркера и выдача всех требуемых выходных сигналов. В данное состояние попадает ФБ, имеющий *R_CHECK= = TRUE* и *M = 0,* или ФБ, уже прошедший *M* раз по петле;

7) состояние *NOT_CHECK*, в которое автомат переходит, если не принят стеком (*R_CHECK = FALSE*). Кроме того, в данном состоянии выдаются выходной сигнал *No_Check*.

Интерфейс ФБ *Stack* состоит из следующих групп параметров:

Входные сигналы:

INIT – инициализация стека;

PUSH – положить значение в стек;

POP – извлечь значение из стека;

CHECK – проверка значения входной переменной *VAR_POP* с верхним значением стека, результат данной проверки выдается в выходной переменной *CHECK_OUT*.

Выходные сигналы:

INTO – сигнал, означающий, что ФБ инициализирован;

OUT – сигнал, сбрасывающий в результате проверки верхнего значения стека;

Входные переменные:

VAR_POP – переменная, с которой происходит сравнение верхнего символа стека;

VALUE – переменная, значение которой добавляется в стек;

Выходные переменные:

OUT_VALUE – переменная – результат проверки входной переменной *VAR_POP* с верхним значением стека (*TRUE* – значения равны, *FALSE* – не равны).

ФБ *Switch* необходим для перенаправления значений переменных, содержащих магазинные символы, в стек, ввиду того, что в среде *NxtStudio* нельзя подсоединить несколько выходных переменных ФБ к одной входной. ФБ *Switch* в зависимости от определенного сигнала проверки (*Check1…Check*$_k$) сопоставляет соответствующий магазинный символ с вершиной стека и, если символы совпали, то записывает необходимую цепочку в магазин.

Интерфейс данного ФБ состоит из следующих групп параметров:

Входные сигналы:

Check1..Check13 – проверка стека от определенного ФБ-перехода;

In – втолкнуть значение в стек;

No – значение в стек не вталкивать;

Выходные сигналы:

CNF – сигнал, означающий проверку верхнего значения в стеке;

OUT – сигнал, по которому извлекается верхнее значение стека и переменная *PUSH_OUT,* которая содержит необходимую цепочку, которая добавляется в стек;

Входные переменные:

POP1..POP3 – переменные (алфавит стека) с которыми сравнивается верхний магазинный символ;

PUSH0..PUSH6 – переменные – цепочки, которые добавляются в стек;

Выходные переменные:

POP_OUT – переменная, значение которой необходимо сравнить с верхним символом стека.

PUSH_OUT – переменная, значение которой вталкивается в стек.

В представленной работе предложен и протестирован метод реализации детерминированных магазинных автоматов на основе ФБ. Направлением дальнейших исследований является разработка метода реализации селектирующих сетей Петри на основе ФБ МЭК 61499.

Библиографический список

1. Vyatkin, V. IEC 61499 Function Blocks for Embedded and Distributed Control Systems Design / V. Vyatkin. – 3-rd edn. – Pittsburgh: Instrumentation Society of America, 2016. – 261 p.

2. Функционально-блочная реализация недетерминированных конечных автоматов / В. Н. Дубинин, И. В. Сенокосов, А. С. Войнов, Д. Н. Дроздов, В. В. Вяткин // Современные информационные технологии : тр. Междунар. науч.-техн. конф. – Пенза, 2017. – Вып. 25. – С. 5–18.

3. Дубинин, В. Н. Использование моделей магазинных автоматов в проектировании технологических процессов сортировки и сборки изделий / В. Н. Дубинин, И. В. Сенокосов, Л. П. Климкина, В. В. Вяткин // Современные технологии в науке и образовании : сб. тр. II Междунар. науч.-техн. и науч.-метод. конф. (СТНО-2017). – Рязань, 2017. – Т. 1. – С. 10–14.

4. Хопкрофт, Дж. Введение в теорию автоматов, языков и вычислений / Дж. Хопкрофт, Р. Мотвани, Дж. Ульман. – М. : Вильямс, 2002. – 528 с.

5. Сайт nxtStudio (nxtControl). – URL: http://www.nxtcontrol.com/

А. С. Бождай, Ю. И. Евсеева, А. А. Гудков

Пензенский государственный университет, Пенза, Россия

ОПТИМИЗАЦИЯ ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА АДАПТИВНЫХ СИСТЕМ С ПОМОЩЬЮ ТЕХНОЛОГИИ ИНТЕЛЛЕКТУАЛЬНОГО АНАЛИЗА ДАННЫХ

Изложена оригинальная технология самооптимизации и пользовательского интерфейса адаптивных информационных систем. В основе предлагаемой технологии лежат алгоритмы интеллектуального анализа данных.

Поскольку создание пользовательского интерфейса программной системы представляет собой задачу, решаемую человеком (разработчиком программного обеспечения или дизайнером), нет абсолютной гарантии того, что разработанный интерфейс будет оптимальным для всех категорий пользователей. Чем выше сложность программной системы, тем сложнее учесть все факторы, связанные с ее предметной областью и функционированием. Определенные закономерности в работе сложной системы могут быть недоступны пониманию разработчиков на этапе проектирования. Такие закономерности, как правило, становятся выявляемыми только после определенного количества времени эксплуатации программного обеспечения [1, 2].

На рис. 1 представлена схема пространства интерфейсов. Как видно из рисунка, пространство интерфейсов организовано иерархическим образом. Сплошными линиями изображены отношения агрегации между элементами, пунктирными – отношения вызова. Примеры отношения вызова – отображение дочернего окна при нажатии на пункт меню, переход по гиперссылке на другую страницу веб-сайта.

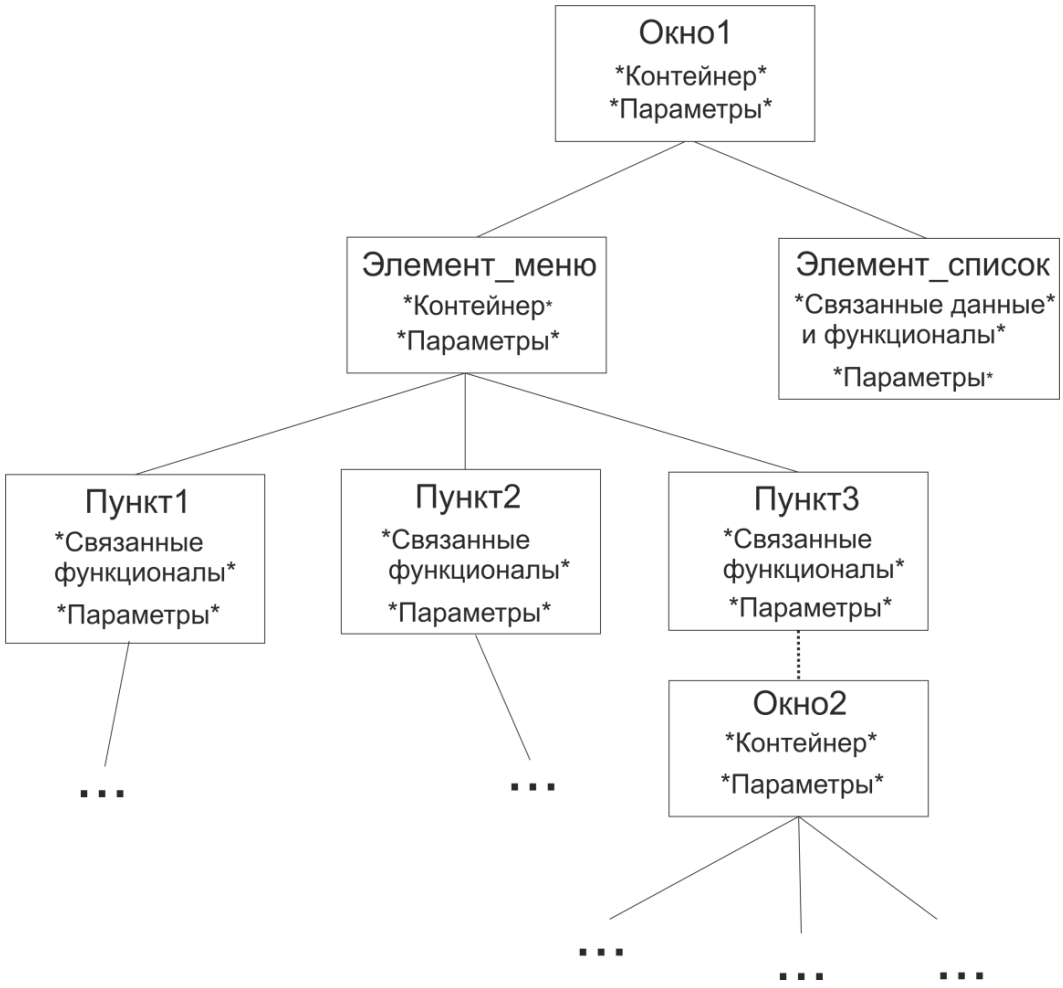


Рис. 1. Пример рис. в тексте доклада

Для оценки оптимальности создаваемых интерфейсов следует воспользоваться вычислением и анализом пользовательских шаблонов навигации. Пользовательский шаблон навигации представляет собой одну из наиболее часто используемых траекторий перемещения пользователя программной системы в пространстве интерфейсов. Для сбора информации о траекториях можно использовать запись пользовательских действий в лог-файл, для анализа и выявления навигационных шаблонов – методы нахождения последовательных шаблонов.

Для анализа поведенческой продукции системы будем использовать алгоритм AprioriALL [3]. На вход алгоритму AprioriALL поступает лог-файл, протоколирующий действия пользователя с системой. В качестве выходных данных выступает список пользовательских шаблонов навигации.

Предлагаемый алгоритм самомодификации пользовательского интерфейса, основанный на технологии интеллектуального анализа данных [4], содержит следующие шаги:

1. Считывание списка исходного набора данных из файла.

2. Формирование списка пользовательских шаблонов навигации помощью алгоритма AprioriALL. Список шаблонов навигации будет иметь вид

$$A = \{A_1, A_2, ..., A_z\},\ z = 1,... f,$$
$$A_i = \{A_{i1}, A_{i2} ..., A_{ik}\},\ A_{ij} = (id_j, idF_j, idD_j, v_j) \forall j = 1,..., k_i,$$

где f – число шаблонов; k_i – число элементов i-го шаблона; $id_j \in P_j$ – идентификатор интерфейса j-го элемента i-го шаблона; $idF_j \in FB_j$ – идентификатор задействованного

в операции над *j*-м интерфейсом *i*-го шаблона функционала; $idD_j \in DB_j$ – идентификатор задействованного в операции над *j*-м интерфейсом *i*-го шаблона объекта данных; $v_j \in V_I$ – связанная с воздействием на *j*-й интерфейс *i*-го шаблона информация (пользовательские действия).

3. Цикл по шаблонам, $s = 1,..., f$.

4. Внутренний цикл по шаблонам, $r = s + 1..., f$.

5. Просмотр очередного шаблона A_r и проверка на то, является ли он схожим с шаблоном A_s . Для определения схожести шаблонов определяются совпадающие в обоих шаблонах подпоследовательности и оценивается порядок их следования. В схожих шаблонах порядок следования идентичных подпоследовательностей должен совпадать. Шаблоны можно считать достаточно схожими, если число элементов, не входящих в идентичные подпоследовательности, не превышает определенного порога.

6. Если A_s и A_r являются схожими шаблонами, то осуществляется их «слияние» – формирование на первом общем для шаблонов интерфейсе-контейнере нового дочернего интерфейса, реализующего совпадающую функциональность шаблонов и использующего общие данные. В процессе выполнения операций, связанных с новым интерфейсом, осуществляется обращение к интерфейсам, входящим в неидентичные подпоследовательности, для получения необходимых данных. Если A_s не является схожим шаблоном, то осуществляется переход к пункту 8.

7. Поиск элементов из совпадающих подпоследовательностей в других шаблонах списка. Если какой-либо элемент не встречается в других шаблонах, то принимается решение о том, что данный элемент является редко используемым. Редко используемый элемент становится скрытым элементом.

8. Завершение внутреннего цикла по шаблонам. Если еще есть непросмотренные шаблоны, переход к пункту 5, если нет – к пункту 9.

9. Если для шаблона A_s не были найдены схожие шаблоны, осуществляется оптимизация интерфейса на основе информации, содержащейся в A_s . На первом интерфейсе-контейнере шаблона формируется новый дочерний интерфейс, реализующий функциональность и работающий с данными всего шаблона.

10. Завершение внешнего цикла по шаблонам.

11. Завершение работы алгоритма.

Предлагаемый алгоритм, разработанный в рамках представлений о рефлексивной адаптации программных систем, позволяет решить 2 основные проблемы, возникающие в процессе создания и эксплуатации пользовательских интерфейсов: проблему недостаточности пространства интерфейсов (отсутствия элементов управления для наиболее часто выполняемых операций) и проблему избыточности (наличие элементов, реализующих «несамостоятельную» функциональность).

Библиографический список

1. Keromytis, A. Characterizing Software Self-healing Systems / A. Keromytis // International Conference on Mathematical Methods, Models, and Architectures for Computer Network Security. – Berlin : Springer, 2007. – P. 22–23.

2. Эванс, Э. Предметно-ориентированное проектирование: структуризация сложных систем / Э. Эванс. – М. : Вильямс, 2011. – 444 с.

3. Анализ данных и процессов / А. Барсегян, М. Куприянов, И. Холод, М. Тесс, С. Елизаров. – СПб. : БХВ-Петербург, 2009. – 314 с.

4. Бутчер, С. Программы лояльности и клубы постоянных клиентов / С. Бутчер. – М. : Вильямс, 2004. – 272 с.

Ю. И. Евсеева

Пензенский государственный университет, Пенза, Россия

МНОГОУРОВНЕВАЯ АРХИТЕКТУРА АДАПТИВНОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ

Рассматриваются проблемы создания адаптивных информационных систем. Помимо разработки специализированных методов синтеза программного обеспечения такого типа, существенной частной проблемой является проблема создания универсальной архитектуры адаптивной системы. Для ее решения в данной статье предложен оригинальный архитектурный шаблон.

В настоящее время среди специалистов отсутствует единое определение понятию адаптивности программного обеспечения. Ситуация такова, что для разных типов программных систем адаптивность может определяться различным образом. Взгляды на реализацию процесса адаптации могут также различаться и в пределах рассмотрения одного класса систем и сильно зависеть от выбранной методологии проектирования [1–3].

Тем не менее, несмотря на кажущиеся существенные различия в определениях и подходах, можно выделить одно общее свойство всех систем, называемых адаптивными [4, 5]: способность к самомодификации (на уровне структуры системы в целом либо на уровне отдельных компонентов).

Любая программная система построена из компонентов различных типов. Хорошим стилем в проектировании программных систем считается минимизация количества зависимостей между отдельными компонентами (модульный подход). Многие совре-менные способы проектирования программного обеспечения (например, сервис-ориентиро-ванная архитектура) предполагают также наличие унифицированных интерфейсов для взаимодействия модулей друг с другом и системой в целом. Как показывает практика, гибкость архитектуры оказывает огромное влияние на способность программы адаптироваться к изменениям и усложнениям. Гибкая архитектура – это, как правило, результат детального моделирования и тщательного выбора проектных решений.

Построение универсального шаблона гибкой архитектуры для абсолютно всех возможных в реализации программных систем вряд ли является выполнимой задачей. Однако, если рассматривать организацию системы только с точки зрения интересующих свойств (например, адаптивности), можно создать архитектурный шаблон, позволяющий проектировать программы, оптимальные именно с позиции рассматриваемого свойства.

На рис. 1 приведена упрощенная архитектурная модель адаптивной системы. Основная идея предлагаемой модели заключается в том, что компоненты каждого из существующих архитектурных уровней будут собственным образом реагировать на необходимость в реорганизации.

Интерфейсный уровень архитектуры содержит объекты-интерфейсы, предназначенные для взаимодействия с конечным пользователем. Уровень данных объединяет используемую в работе системы информацию (например, информацию из баз данных). Уровень функционалов включает в себя 3 уровня: уровень функционалов предметной области (программные операции, выполняемые отдельными объектами предметной области); уровень операционных функционалов (распределяют задачи между объектами предметной области); уровень инфраструктурных функционалов (обеспечивают непосредственную техническую поддержку для верхних уровней).



Рис. 1. Обобщенная архитектура адаптивной информационной системы

Наибольший интерес представляют модели предметной области. Благодаря специализированным предметно-ориентированным языкам, конечный пользователь системы видит их как цельные понятия предметной области. Однако окружающей архитектурной средой эти понятия представляют собой комбинацию компонентов 2 смежных уровней – уровня данных и уровня функционалов. С этой точки зрения, модели предметной области являются аналогами классов в объектно-ориентированном проектировании, однако в предлагаемой модели разделение между функциональным уровнем и уровнем данных акцентируется и в том случае, когда речь идет о цельном понятии предметной области. Это связано с тем фактом, что функционалы и данные различным образом реагируют на необходимость в изменениях.

Для рассмотренной ориентированной на самомодификацию архитектуры механизм адаптации определяется следующим соотношением:

$$S_A = A(S, L(S, V)), \tag{1}$$

где S_A – модифицированная структура системы; A – функция анализа поведенческой продукции системы и модификации ее первоначальной структуры; L – функция вычисления поведенческой продукции системы; S – исходная структура системы (включает совокупность объектов адаптируемых уровней архитектуры); V – совокупность внешних воздействий на адаптируемые объекты.

Применительно к S также справедливо

$$S = \{I, D, F_A, M\},$$

где I – множество объектов уровня интерфейсов; D – множество объектов уровня данных; $F_A = F \setminus F_I = \{F_D, F_O\}$ – множество адаптируемых функционалов; M – множество объектов предметной области. К адаптируемым функционалам относятся функционалы уровня предметной области (F_D) и функционалы операционного уровня (F_O). Функционалы уровня инфраструктуры (F_I) не являются адаптируемыми объектами.

Множество объектов предметной области представляет собой неориентированный гиперграф

$$M = \{W_D, W_F, E\},$$

где W_D – множество вершин, отображающих используемые структуры данных (точки на уровне данных рис. 2); W_F – множество вершин, отображающих функционалы предметной области; E – семейство непустых подмножеств множества $W_D \bigcup W_F$ (цельных объектов предметной области).

Следует отметить, что адаптация уровня целостных объектов предметной области не сводима к адаптации отдельных компонентов этих объектов (функционалов и данных). Помимо оптимизации функционалов и данных, она включает в себя также изменение структуры целостного объекта. Изменение структуры объектов реализуется через операции над гиперграфом *M*.

Библиографический список

1. Keromytis, A. D. Characterizing Software Self-healing Systems / A. D. Keromytis // Computer Network Security. – Berlin : Springer, 2007. – P. 22–33.

2. Shin, M. E. Connector-Based Self-Healing Mechanism for Components of a Reliable System / M. E. Shin, D. Cooke // Workshop on the Design and Evolution of Autonomic Application Software (DEAS 2005). – New Yourk : ACM, 2005. – P. 1–7.

3. Smart Sparrow. – URL: https://www.smartsparrow.com (дата обращения: 22.09.2017).

4. Fuzzy control-based software self-adaptation: A case study in mission critical systems / Q. Yang, J. Lü, J. Xing, X. Tao, H. Hu, Y. Zou // IEEE 35th Annual Computer Software and Applications Conference Workshops (COMPSACW). – Washington : IEEE Computer Society, 2011. – P. 13–18.

5. Liu, L. Requirements cybernetics: elicitation based on user behavioural data / L. Liu, Q. Zhou, J. Liu, Z. Cao // J. Syst. Software. – Amsterdam : Elsevier, 2016.

И. Ю. Балашова, С. В. Шибанов, А. Ю. Афонин

Пензенский государственный университет, Пенза, Россия

ПРИМЕНЕНИЕ ТЕХНОЛОГИИ БЛОКЧЕЙН В ПОСТРОЕНИИ ДЕЦЕНТРАЛИЗОВАННЫХ РАСПРЕДЕЛЕННЫХ ИНФОРМАЦИОННЫХ СИСТЕМ

Выделены группы блокчейнов в соответствии с доступом к данным. Описаны преимущества применения эксклюзивных блокчейнов в построении децентрализованных распределенных информационных систем.

Технология блокчейна представляет собой совокупность инновационных решений в области организации, хранения и обработки распределенных данных. Блокчейн – это распределенная база данных для обработки транзакций [1]. Одно из главных преимуществ блокчейна по сравнению с другими моделями распределенных баз данных – интеграция обработки данных, обеспечения корректности и безопасности в единый протокол, реализуемый алгоритмически и минимизирующий человеческий фактор.

Блокчейн-система работает по следующим правилам.

1) новые транзакции рассылаются всем узлам;

2) каждый узел объединяет пришедшие транзакции в блок;

3) каждый узел пытается подобрать хэш блока, удовлетворяющий текущей сложности;

4) как только такой хэш найден, блок отправляется в сеть;

5) узлы принимают этот блок, только если все транзакции в нем корректны и не используют уже потраченные средства;

6) свое согласие с новыми данными узлы выражают, начиная работать над следующим блоком и используя хэш предыдущего в качестве новых исходных данных.

Основным принципом функционирования технологии блокчейна является прозрачность совершаемых операций с невозможностью их изменения лицами, не имеющими к ней санкционированного доступа. Все транзакции в блокчейне хранятся в едином реестре. Поскольку транзакции полностью упорядочены по времени, текущее состояние системы

определяется исключительно этим реестром транзакций. Хранение полной истории изменений состояния системы имеет свои преимущества, например, возможность определить состояние системы в произвольный момент времени, просто «проиграв» заново соответствующие транзакции.

В идеальном случае, обработка транзакций в рамках блокчейн-технологии удовлетворяет следующим свойствам [2]:

- транзакции согласованы с текущим состоянием системы;
- транзакции авторизованы;
- транзакции неизменяемы: после того, как транзакция записана в реестр, ее должно быть невозможно изменить;
- транзакции конечны: после того, как транзакция записана в реестр, ее будет невозможно удалить оттуда;
- устойчивость к цензуре: если транзакция удовлетворяет всем правилам блокчейна, она должна быть, в конце концов, в него добавлена.

Соответствие текущему состоянию системы удовлетворяется за счет проверки транзакции совместно с этим состоянием, хранящимся в защищенной от злоумышленников памяти. Поскольку текущее состояние системы можно восстановить при помощи блокчейна, предположение защищенности не сужает безопасность системы в целом. Это предположение вводит ограничение на блокчейн, которое заключается в организации хранения транзакций таким образом, чтобы надежная верификация транзакций занимала не слишком много времени.

Проблема авторизации решается за счет использования криптографии с открытым ключом. Каждому пользователю системы выдается пара из секретного и открытого ключа; открытый ключ может быть без проблем опубликован для определения цифровой личности пользователя, так как секретный ключ невозможно вывести из открытого. Использование цифровых подписей решает не только проблему авторизации, но также и проблему изменяемости транзакций. Если цифровые подписи используются для всех транзакций в блокчейне, злоумышленник, получивший внутренний доступ к системе, не может изменить эти транзакции.

Неизменяемость и конечность транзакций в системе на основе блокчейна достигается при помощи разделения транзакций в блоки, упорядоченные во времени, и расчете криптографической хэш-функции для каждого из блоков.

При использовании блокчейн-технологии состояние системы, которое является общим для всех узлов – это сам блокчейн; новые транзакции изначально не принадлежат ни одному из блоков, т.е. являются неподтвержденными. Протокол блокчейна определяет правила по созданию новых блоков; когда новый блок создан, он распространяется по сети, и множество неподтвержденных транзакций на всех узлах системы меняется в соответствии с новым состоянием (в частности, противоречащие новому состоянию транзакции удаляются). Это правило гарантирует, что, в то время как неподтвержденные транзакции могут противоречить друг другу, транзакции в блокчейне всегда соответствуют протоколу его построения. Адекватный протокол минимизирует вероятность того, что несколько узлов одновременно добавят «свои» блоки в блокчейн. Например, если в системе существует ограниченное количество известных узлов, они могут создавать блоки по очереди с интервалом, который в несколько раз превышает интервал распространения блока по системе.

Выделяют четыре вида блокчейнов в соответствии с доступом к данным [4]:

- открытый блокчейн – это блокчейн, в котором не существует ограничений на чтение данных блоков (при этом данные могут быть зашифрованы) и ограничений на отсылку транзакций для включения в блокчейн;
- закрытый блокчейн – это блокчейн, в котором прямой доступ к данным и к отправке транзакций ограничен определенным узким кругом организаций;
- общедоступный (инклюзивный) блокчейн – это блокчейн, в котором не существует ограничений на личность обработчиков транзакций (т.е., пользователей, которые могут создавать блоки транзакций);

• эксклюзивный блокчейн – это блокчейн, в котором обработка транзакций осуществляется определенным списком субъектов с установленными личностями.

В эксклюзивных блокчейнах существует несколько уровней доступа к данным блокчейна, например:

• чтение транзакций из блокчейна, возможно, с определенными ограничениями (например, у клиента может быть доступ только к транзакциям, которые затрагивают его);

• предложение транзакций для включения в блокчейн;

• создание новых блоков транзакций и добавление блоков в блокчейн.

В то время как третий уровень доступа в эксклюзивных блокчейнах предоставляется ограниченному кругу учреждений (например, лицензированным обработчикам транзакций), доступ к блокчейну не обязан ограничиваться этими учреждениями. Напротив, учреждения, управляющие эксклюзивным блокчейном, теоретически могут

• предоставить доступ на чтение транзакций и заголовков блоков (возможно, ограниченный) для клиентов с целью обеспечения технологичного, прозрачного и надежного способа проверки сохранности средств клиентов;

• предоставить полный доступ на чтение данных блокчейна для регулирующих учреждений для соответствия законам;

• предоставить всем лицам и учреждениям к доступом к данным исчерпывающее описание протокола блокчейна, который включает в себя объяснение возможных взаимодействий с данными блокчейна.

Эти шаги могут облегчить независимый аудит и проверку целостности данных блокчейна, например, регулирующими учреждениями.

Если база данных блокчейна полностью непрозрачна для клиентов (т.е. клиенты не имеют доступ к данным блокчейна), то безопасность блокчейна уменьшается. Поскольку встроенный протокол авторизации транзакций является одним из ключевых моментов блокчейн-технологии, его игнорирование в пользу централизованных решений может отрицательно сказаться на защищенности системы. Кроме того, так как транзакции в этом случае доступны ограниченному числу компьютеров, возникают риски вмешательства человеческого фактора в функционирование этих компьютеров, причем клиенты не смогут выявить это вмешательство. Таким образом, непрозрачный дизайн блокчейнов по своей сути подрывает два ключевых аспекта блокчейн-технологии: децентрализацию и опору на алгоритмически заложенные правила обработки транзакций без вмешательства человека. В целом, любое взаимодействие с данными блокчейна в обход его протокола вводит в систему уязвимости.

Блокчейн соединяет в себе преимущества нескольких технологий: децентрализованных сетей, распределенных баз данных и криптографических алгоритмов. Главным достоинством технологии блокчейн является предельная достоверность и защищенность от изменения записей, что достигается за счет применения специальных криптографических алгоритмов и азмещения полной версии базы данных на всех узлах сети. Дополнительным преимуществом блокчейна является возможность реализовать в его среде обмен любыми электронными активами без посредников с сохранением приватности.

Юридические и технические ограничения, риски мошенничества и угрозы безопасности делают невозможным применение открытых блокчейнов в распределенных информационных системах финансовых и государственных учреждений. Эксклюзивные блокчейны привлекательны для учреждений, работающих с реестрами и финансовыми системами учета, поскольку большинство систем законодательства предусматривает необходимость регистрации обработчиков транзакций. Эти блокчейны могут сформировать более контролируемую и прогнозируемую среду, чем общедоступные блокчейны.

Библиографический список

1. Siraj Raval, Decentralized Applications: Harnessing Bitcoin’s Blockchain Technology. – Published by O’Reilly Media, Inc., 2016. – 104 p.

2. Distributed Ledger Technology: beyond block chain. A report by the UK Government Chief Scientific Adviser. Available at. – URL: https://www.gov.uk/government/uploads/system/uploads/attach-ment _data/file/492972/gs-16-1-distributed-ledger-technology.pdf (accessed 4 April 2017).

3. Unlocking economic advantage with blockchain: a guide for asset managers. Available at: – URL: http://www.oliverwyman.com/our-expertise/insights/2016/jul/unlocking-economic-advantage-with-blockchain.html (accessed 7 April 2017).

Д. В. Ломизов, А. В. Дубравин

Пензенский государственный университет, Пенза, Россия

АВТОМАТИЗАЦИЯ ТЕСТИРОВАНИЯ ПРОГРАММНО-АППАРАТНЫХ КОМПЛЕКСОВ

Рассмотрен вопрос автоматизации процесса тестирования программно-аппаратных комплексов. В качестве средства автоматизации описан модуль на базе платформы Arduino.

В своей работе в отделе тестирования ООО НТП «Криптософт» мы часто сталкиваемся с задачей проверки функционирования программно-аппаратных комплексов. Многие из них реализованы на основе аппаратных средств, имеющих собственную консоль управления. Тестирование функционала данных комплексов, выполняющегося под управлением операционной системы можно достаточно легко автоматизировать, используя как мощные интегрированные средства тестирования (Ranorex, TestComplete), так и отдельные платные или бесплатные фреймворки. В то же время, проверка функционирования консолей управления долгое время оставалась задачей исключительно для ручного тестирования.

Таким образом, актуальным является создание средства, позволяющего автоматизировать данный вид работ частично или полностью.

Рассмотрим преимущества и недостатки автоматического тестирования:

- повторяемость тестов. Тест будет выполняться абсолютно однообразно раз за разом, что позволит во-первых без лишних усилий проверить исправление найденной в процессе такого тестирования ошибки, а во-вторых создать постоянно расширяемый набор тестов для регрессионного тестирования;
- высвобождение рабочего времени сотрудников для проведения тестов, требующих участия человека, например оценка эргономики интерфейса, выработка рекомендаций для разработчиков и т.п.
- возможность автоматически собирать и анализировать информацию о реакции системы на тест путем вывода и анализа логов;
- возможность проводить длительные тесты без участия человека, в том числе в нерабочее время, что повышает эффективность использования оборудования и снижает общие временные затраты на проведение полного цикла тестирования комплекса;
- возможность проведения интенсивных нагрузочных тестов, в том числе нагрузочное тестирование отдельных узлов аппаратуры, к примеру – контроллера Ethernet или USB;
- единожды созданные тесты могут многократно использоваться повторно, как для тестирования текущего проекта, так и для вновь возникающих задач, как в адаптированном, так и в исходном виде. При накоплении достаточно большого массива тестов, время на подготовку к тестированию нового устройства будет сокращаться за счет использования подобных заготовок;

- при проведении автоматического тестирования могут быть пропущены ошибки, на которые мог бы «попутно» обратить внимание тестировщик и, выполнив дополнительные действия, локализовать ошибку;
- для тестирования аппаратных средств необходима разработка как аппаратных же модулей, взаимодействующих с тестируемым устройством, так и программного обеспечения, использующегося для управления процессом тестирования, сбора и анализа информации, полученной в результате. Это влечет за собой дополнительные финансовые и временные затраты, а также увеличивает требования к необходимой квалификации сотрудников;
- при анализе существующих средств тестирования не найдено готовых бесплатных или коммерческих решений, направленных на тестирование аппаратной составляющей при функционировании до загрузки операционной системы.

Сформулируем требования, предъявляемые к необходимому нам средству автоматизированного тестирования.

При исследовании функционирования аппаратных средств, нас интересует реакция на следующие виды воздействий:

- срабатывание различных внешних датчиков;
- аппаратные сбои в работе ПК;
- действия пользователя по вводу информации.

Кроме того, важной задачей является получение информации о действиях, выполняемых программным обеспечением комплекса.

На этапах отладки и тестирования получение и журналирование информации о выполняемых ПО комплекса действиях осуществляется путем вывода данной информации в serial-порт, которых на платах, входящих в состав комплекса, может быть несколько. Данные с этих портов поступают на ПК тестировщика, где хранятся и обрабатываются. При этом в процессе разработки аппаратных прошивок контроллеров возможно изменять и расширять состав выводимой информации.

Для реализации вышеперечисленных требований было использовано устройство на основе платформы Arduino. Управление данным контроллером осуществляется с ПК тестировщика через serial-интерфейс. Обратная связь с тестируемым устройством в настоящее время отсутствует. Информация о выполнении тестов получается напрямую с тестируемого устройства на ПК тестировщика.

Рассмотрим основные функции разработанного устройства.

Сигналы внешних датчиков, используемых в нашей аппаратуре, чаще всего сводятся к замкнутому/разомкнутому контакту, либо наличию логического уровня на выходе датчика. При тестировании к разъемам датчиков подключаются шлейфы, на которые подается сигнал, инициированный контроллером. Предусмотрены команды «замкнуть контакты», «разомкнуть контакты», «послать 0», «послать 1», а также имитация «дребезга контактов», при которой контакты шлейфа многократно замыкаются и размыкаются через случайные промежутки времени в интервале от 20 до 40 миллисекунд в течение заданного отрезка времени.

Подобным образом реализована и эмуляция аппаратных сбоев. Контроллер управляет прерывателями, включенными в разрыв кабелей питания и/или информационных шлейфов. Реализованы команды «замкнуть цепь», «разомкнуть цепь».

Эмуляция действий пользователя по вводу информации реализована следующим образом. USB-контроллер на плате Arduino прошит модифицированной прошивкой, которая позволяет при подключении к ПК, на котором установлен тестируемый программно-аппаратный комплекс, эмулировать USB HID-устройство. Для управления виртуальной клавиатурой используются команды «клавиша нажата», «клавиша отпущена» (эмулируется соответственно нажатие и отпускание клавиши), а так же «клик» по клавише, при котором эмулируется кратковременное (на случайный интервал от 30 до 75 миллисекунд) нажатие и последующее отпускание клавиши.

Использование разработанного устройства позволяет заметно сократить время, затрачиваемое сотрудником на проведение рутинных проверок и оставляет возможность тестировать комплекс в «свободном поиске».

В качестве дальнейшего развития предполагается дополнить устройство USB-хабом с управляемым подключением/отключением к хосту включенных в хаб устройств.

Библиографический список

1. Борисова, Т. М. Особенности автоматизации тестирования программно-аппаратных СЗИ / Т. М. Борисова. – URL: http://www.okbsapr.ru/borisova_2013_1.html

О. С Дорофеева, Б. В. Казаков, И. А. Казакова

Пензенский государственный университет, Пенза, Россия

ПРИМЕНЕНИЕ ОБЪЕКТНО-ОРИЕНТИРОВАННОГО ПОДХОДА ПРИ РАЗРАБОТКЕ АВТОМАТНОЙ МОДЕЛИ ПРОГРАММНЫХ СИСТЕМ

Рассматриваются вопросы применения объектно-ориентированного подхода при разработке автоматной модели программных систем.

В формулировке цели разработки программной системы участвуют предметы и понятия реального мира, имеющие отношение к разрабатываемой программной системе. При объектно-ориентированном подходе эти предметы и понятия заменяются их моделями, т.е. определенными формальными конструкциями, представляющими их в программной системе.

Формальный характер моделей позволяет получить формальную модель разрабатываемой программной системы как композицию формальных моделей ее компонентов.

Таким образом, объектно-ориентированный подход помогает справиться с такими сложными проблемами, как

– уменьшение сложности программного обеспечения;

– повышение надежности программного обеспечения;

– обеспечение возможности модификации отдельных компонентов программного обеспечения без изменения остальных его компонентов;

– обеспечение возможности повторного использования отдельных компонентов программного обеспечения.

Применение объектно-ориентированного подхода позволяет разрабатывать хорошо структурированные, надежные в эксплуатации, достаточно просто модифицируемые программные системы.

Автоматная модель программной системы представляет собой граф, вершины которого называются состояниями и изображают ситуации (или классы ситуаций), в которых может находиться система в различные моменты времени. Ребра графа могут иметь метки, изображающие действия, которые может исполнять система.

Функционирование системы изображается в данной модели переходами по ребрам графа от одного состояния к другому. Если проходимое ребро имеет метку, то эта метка изображает действие системы, исполняемое при переходе от состояния в начале ребра к состоянию в его конце.

Построение модели системы начинается с анализа предметной области. Стандартными средствами [1] строится статическая модель системы и, согласно автоматному подходу, в статической модели выделяются объекты и управляющие ими автоматы.

Автоматная методология описывает динамическую модель системы с помощью так называемой *модели управления*. Построение этой модели состоит из следующих этапов:

– в статической модели выделяются автоматы, все возможные объекты, являющиеся источниками событий, и объекты управления, которые также могут являться источниками входных воздействий;

– все объекты компонуются на схеме связей автомата, которая представляется с помощью диаграммы классов *UML*;

– каждое входное и выходное воздействие реализуется вручную в соответствии с его функциональностью;

– для каждого автомата описывается поток управления, представленный в виде графа переходов, заданного диаграммой состояний *UML*, в которой на дугах указываются события, входные и выходные воздействия;

– с использованием подходящего инструмента разработки программного обеспечения диаграмма классов и графы переходов могут быть автоматически преобразованы в исходный код на целевом языке или интерпретированы специальным интерпретатором.

Основными составляющими функциональной модели являются диаграммы потоков данных [2], которые показывают потоки значений от внешних входов через операции и внутренние хранилища данных к внешним выходам.

Помимо диаграмм потоков данных, функциональная модель описывает смысл операций объектной модели и действий динамической модели, а также ограничения на объектную модель.

Согласно функциональной модели, все операции можно разделить на три категории: *запросы*, *действия* и *активности*. Запросом называется операция без побочных эффектов над видимым извне объекта его состоянием. Примером запроса может служить входное воздействие автоматной модели. Действием называется операция, имеющая побочные эффекты, которые могут влиять на целевой объект и на другие объекты системы, которые достижимы из целевого объекта. Примером действия является выходное воздействие.

Все операции в модели должны быть специфицированы. Обычно спецификация операции содержит ее характеристики (имя, количество, порядок и типы параметров, количество, порядок и типы возвращаемых значений) и описание ее эффекта (действия, преобразования). Операции автоматной модели не имеют параметров. Выходные воздействия к тому же не имеют возвращаемых значений. Однако и они, и входные воздействия, могут получать и изменять данные в памяти автомата. Запрашиваемые действием элементы данных называются *входными данными* (или параметрами). Сохраняемые в память автомата данные называются *выходными данными действия*. Кроме того, события в автоматной модели также несут с собой данные – свои параметры, которые называются *выходными данными события*.

Итак, для автоматной модели целесообразно описывать имена операций (действий и событий), входные и выходные данные, а также эффект операций. Для описания эффекта операций в функциональной модели могут использоваться:

– математические формулы;

– табличные функции: таблицы, сопоставляющие выходные значения входным;

– уравнения, связывающие входные и выходные значения;

– аксиоматическое определение пред- и постусловий;

– таблицы принятия решений;

– псевдокод;

– естественный язык.

Таким образом, для спецификации воздействий и событий автоматной модели достаточно указания списков входных и выходных данных, а также описания эффекта этих действий на естественном языке.

Библиографический список

1. Гайсарян, С. С. Объектно-ориентированные технологии проектирования прикладных программных систем / С. С. Гайсарян. – URL: http://www.citforum.ru/programming/oop_rsis/index.shtml

2. Казаков, Б. В. Объектно-ориентированная технология разработки программных систем и диаграммы потоков данных / Б. В. Казаков, И. А. Казакова // Информационные технологии в науке и образовании. Проблемы и перспективы : сб. науч. ст. IV Ежегодной межвуз. науч.-практ. конф. (МНПК – 2017), (г. Пенза, 15 марта 2017 г.). – Пенза : Изд-во ПГУ, 2017. – С. 121–124.

А. А. Доломанов

Пензенский государственный университет, Пенза, Россия

ДИСТАНЦИОННОЕ УПРАВЛЕНИЕ СРЕДСТВАМИ ИЗМЕРЕНИЙ НА ПРИМЕРЕ SMB100 ROHDE&SCHAWZ

Рассматриваются вопросы автоматизации поверки СИ, что исключает человеческий фактор, ускоряет производство и автоматизирует заполнение документации, при этом не исключая режим ручного управления.

Актуальность автоматизации поверки средств измерений (СИ) заключается в возможности исключить человеческий фактор и ускорить производство и автоматизирует заполнение документации, при этом не исключая режим ручного управления. Данная статья содержит информацию:

– о настройке генератора сигналов для работы в режиме дистанционного управления;

– об общих принципах дистанционного управления программируемыми измерительными приборами, включая описание структуры команд и синтаксиса применительно к стандарту SCPI, а так же описание выполнения команд и содержимого регистров состояния.

Прибор имеет следующие интерфейсы для дистанционного управления:

– интерфейс шины IEC/IEEE, соответствующий стандарту IEC 625.1/EEE 488.2;

– сетевой интерфейс LAN: в сетевой карте используется протокол 100МГц Ethernet IEEE 802.3u; этот протокол основывается на стандарте VXI-11.

– интерфейс USB: интерфейс B-типа (USB-устройство) используется для дистанционного управления.

Разъемы дистанционного управления расположены на задней стенке прибора, контроллер дистанционного управления может подключаться к ним на прямую или через локальную сеть (LAN).

Необходимым условием для дистанционного управления прибором по сетевому или USB-интерфейсу является установка VISA-библиотек на стороне контроллера. VISA – это стандартизированная библиотека программируемого интерфейса, содержащая функции ввода и вывода для взаимодействия с приборами.

Для дистанционного управления используются команды стандарта SCPI (Standard Commands for Programmable Instruments), имеющие вид сообщений. Команды, не входящие в стандарт SCPI, тем не менее, соответствуют синтаксическим правилам этого стандарта. Прибор поддерживает версию стандарта SCPI от 1999 г. Сам же стандарт SCPI основывается на стандарте IEEE 488.2 и предназначен для стандартизации команд конкретных устройств, обработки ошибок и регистров состояния.

Изложенная ниже простая и короткая последовательность действий позволит быстро привести прибор в рабочее состояние и установить его основные функции. При этом

адрес прибора на шине IEC/IEEE (при выпуске прибора установлен адрес 28) менять не следует.

– соединить прибор с контроллером через кабель шины IEC/IEEE, затем включить их;

– составить и запустить на контроллере следующую программу:

```
CALL IBFIND (“DEV1”,                ‘Открыть порт для обмена данных с прибором
generator %)
CALL IBPAD (generator %, 28)        ‘Сообщить контроллеру адрес прибора
CALL IBWRT (generator %,            ‘Сброс прибора
“*RST; *CLS”)
CALL IBWRT (generator %,            ‘Установить частоту 50 МГц
“FREQ 50MHz ”)
CALL IBWRT (generator %,            ‘Установить уровень сигнала выходе – 7,3 дБм
“POW – 7. 3dBm”)
CALL IBWRT (generator %,            ‘Включить выход ВЧ
“OUTP : STAT ON”)
CALL IBWRT (generator %,            ‘Установить источник амплитудной модуляции LFGEN
“AM : SOUR INT”)
CALL IBWRT (generator %,            ‘Установить частоту модуляции 15 кГц
“LFO : FREQ 15kHz”)
CALL IBWRT (generator %,            ‘Установить глубину амплитудной модуляции 30 %
“AM 30PCT”)
CALL IBWRT (generator %,            ‘ Включить амплитудную модуляцию
“AM : STAT ON”)
```

В результате выполнения этих команд на ВЧ-выход генератора сигналов будет выводиться амплитудно-модулированный сигнал.

Чтобы вернуться в режим ручного управления, нажмите клавишу LOCAL на передней панели прибора.

При включении питания прибор всегда устанавливается в режим ручного управления, при этом управления может производится или органами управления на передней панели прибора, или мышью и внешней клавиатурой.

При дистанционном управление по шине IEC/EEE прибор устанавливается в режим дистанционного управления (состояние REMOTE) по средствам любой адресной команды.

При дистанционном управлении через сеть Ethernet режим дистанционного управления посредствам команды не устанавливается. Прибор должен быть предварительно установлен именно в состояние REMOTE, например, посылкой команды &GTR (перейти в состояние REMOTE).

В состоянии REMOTE управление прибором органами управления на передней панели прибора или мышью и внешней клавиатурой невозможны. Однако возможно открытие меню (например, для проверки настроек). Кнопки и области настроек не выделены и активированы быть не могут. В строке состояния отображается сообщение REMOTE.

Прибор остается в режиме дистанционного управления REMOTE до тех пор, пока не будет задан режим ручного управления – вручную (клавишей LOCAL на передней панели прибора) или командой &GTR через интерфейс дистанционного управления.

Переход от дистанционного управления прибором к ручному и наоборот каких-либо других настроек прибора не затрагивает. В режиме дистанционного управления клавиши на передней панели, а также мышь и внешняя клавиатура (если они подключены) могут быть заблокированы командой SYST : KLOC ON. В этом случае меню открыть нельзя, а переход не ручное управление возможен только посредствам соответствующей команды дистанционного управления. Клавиша LOCAL в этом случае не действует.

Действие этой клавиши в отдельности может быть заблокировано командой &LLO.

Для осуществления дистанционного управления по интерфейсу USB компьютер и прибор должны быть соединены через USB-интерфейс типа B.

Для осуществления USB-соединения на контроллере должна быть установлена библиотека VISA. При установлении USB-соединения VISA автоматически обнаружит и сконфигурирует генератор R&S SMB. При этом ввода адресной строки и установки отдельного драйвера не требуется.

Используемая адресная строка USB:

USB::<vendor Id>::<product Id>::<серийный номер>

Для Rohde&Schwarz строка “vendor Id” – это 0x0AAD, строка “product Id” – 0x54, а серийный номер – это индивидуальный серийный номер, нанесенный на заднюю стенку прибора.

Пример:

“USB::0x0AAD::0x0054::100001::INSTR”

Таким образом, представленными выше вариантами дистанционного управления СИ, оптимальными для вывода., можно реализовать осуществлять автоматическую поверку и калибровку новых источников питания, поддерживающих дистанционное управление. Имея протоколы поверки, с внесенными данными о пределах погрешности, достаточно всего лишь подключить современный источник питания к электронной нагрузке (АКИП-1303) и запустить программу дистанционного управления. Оба средства измерения подключаются к ДУ через ПК посредствам одного из вариантов представленных выше, т.е. по USB-интерфейсу либо по шине IEC/IEEE. Программа автоматически проводит поверку, используя пределы погрешности, с последующим заключением о поверке/калибровке и выводом на экран для печати протокола поверки. Что исключает человеческий фактор, ускоряет производство и автоматизирует заполнение документации, но не исключает ручного управления.

В. А. Калашников, К. Д. Чиркин

Пензенский государственный университет, Пенза, Россия

РЕАЛИЗАЦИЯ СИСТЕМЫ ТРАНСФОРМАЦИИ БУЛЕВЫХ ФОРМУЛ НА ЯЗЫКЕ PROLOG

Рассматриваются принципы построения системы трансформации и упрощения формул булевой алгебры через основные тождества с помощью средств логического программирования Prolog. Подчеркиваются преимущества языка Prolog по отношению к традиционным языкам программирования при решении некоторых задач. Приводятся примеры работы программы преобразования булевых формул.

При проектировании интеллектуальных систем программист сталкивается с рядом сложностей, возникающих в связи с использованием императивного метода программирования. При традиционном программировании проблема представлена через описание метода ее решения непосредственно процедурно-ориентированным языком. Другими словами, *решением* служит описание алгоритма действий, и *решение* всецело зависит от точности описания этого алгоритма.

Язык Prolog сосредоточен вокруг небольшого набора основных механизмов: сопоставление с образцом, древовидное представление структур данных и автоматический перебор с возвратами. Тем самым Prolog избавляет программиста от утомительного написания сложных алгоритмов анализа и обработки информации.

Prolog является декларативным языком программирования: логика программы выражается в терминах отношений, представленных в виде фактов и правил. Для того чтобы

инициировать вычисления, выполняется специальный запрос к базе знаний, на которые система логического программирования генерирует ответы «истина» и «ложь». Для обобщенных запросов с переменными в качестве аргументов созданная система Prolog выводит конкретные данные в подтверждение истинности обобщенных сведений и правил вывода.

В данной статье представлена программа, осуществляющая символьное преобразование формул булевой алгебры. Программа использует основные законы булевой алгебры. Пример трансформации выражения с поэтапным выводом дан на рис. 1.

```
[2] 20 ?- exp_transform((x v y) =>((~(x)=>y)& ~(~(x)&y))).

x v y=> (~x=>y)& ~ (~x&y)
~ (x v y)v (~x=>y)& ~ (~x&y)
~x& ~y v (~x=>y)& ~ (~x&y)
~x& ~y v (~ (~x)v y)& ~ (~x&y)
~x& ~y v (x v y)& ~ (~x&y)
~x& ~y v (x v y)& (~ (~x)v~y)
~x& ~y v (x v y)& (x v ~y)
~x& ~y v x v y& ~y
~x& ~y v x v false
~x& ~y v x
~y v x
true.
```

Рис. 1. Пример работы программы преобразователя

При получении выражения от пользователя программа обращается к базе фактов, являющейся набором основных свойств и тождеств булевой алгебры. Цикл работы программы представлен следующим образом: механизм Prolog сопоставляет входное выражение с образцом и в случае успеха выводит преобразованное выражение. В новом цикл посылается уже преобразованное выражение. Цикл длится до тех пор, пока Prolog не сможет найти соответствие в базе фактов. Преобразование основывается на принципе «от большего к меньшему», для этого используются удаление тавтологии и законы поглощения и склеивания. Блок-схема алгоритма трансформации представлена на рис. 2.

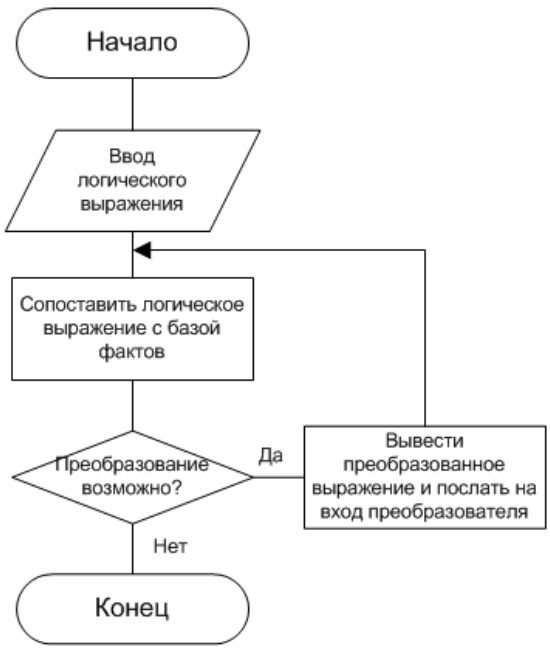


Рис. 2. Блок-схема алгоритма трансформации булевых формул

Программа дает возможность сравнить две любые булевые формулы между собой. Перед тем как сравнить, программа пытается преобразовать выражения в простую форму. Когда преобразование становится невозможным, программа начинает сравнение, опираясь на базу фактов, которая состоит из основных тождеств булевой алгебры. Результат работы программы представлен на рис. 3.

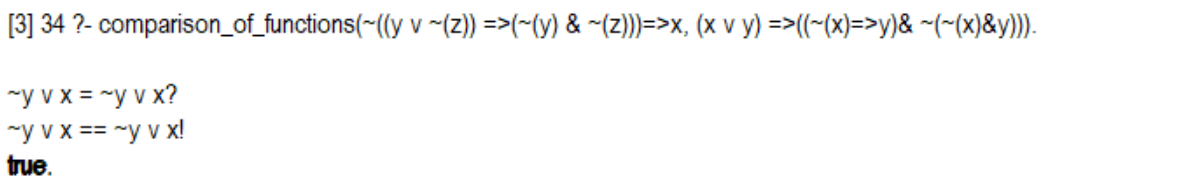


Рис. 3. Пример работы программы преобразования и сравнения

От программиста не требуется написания сложного алгоритма, по которому выражение разделяется на подвыражения, которые сопоставляются по образцу, и в случае соответствия выполняется преобразование. Prolog делает это сам благодаря некоторым особенностям работы его механизмов.

В качестве наглядного примера представлен участок кода программы преобразователя для демонстрации работы Prolog-машины.

```
exp_transform(Term, Result_Term):-
      transform(Term, Result_Term),!,
      nl, write(Term),
      exp_transform(Result_Term, New_Result_Term).

exp_transform(Term, _ ):-
      nl, write(Term).
```

Программа имеет два одноименных предиката – это своего рода аналог If…Then конструкции. Первый предикат осуществляет трансформацию выражения, второй выводит финальное выражение. Механизм Prolog читает код программы по порядку и при обращении к предикату пытается доказать первый вариант в списке. Если Prolog не может достичь цели (трансформация не возможна), то он начинает поиск других вариантов в коде программы. Необходимо подчеркнуть, что переменные в языке Prolog могут хранить любую информацию, в том числе сложные выражения.

В качестве наглядного примера представлен участок кода базы фактов программы преобразователя для демонстрации работы Prolog-машины.

```
      transform((X v Z) & Z, Z).
      transform(X & Y, X1 & Y):-
            transform(X, X1).
```

Пользователь отправляет в качестве аргумента выражение (a v b) & b, после этого Prolog сопоставляет выражение с образцом. В первом случае переменная X = a , Z = b. Если первый факт не будет доказан, то Prolog перейдет ко второму варианту и разделит выражение на подвыражение с помощью рекурсивного вызова предиката. В нем X = (a v b), Y = b. Из-за этих особенностей механизмов языка Prolog код программы, в сравнении с традиционными языками, занимает малый объем.

Библиографический список

1. Клоксин, У. Программирование на языке Пролог / У. Клоксин, К. Меллиш. – URL: https://www.e-reading.club/book.php?book=96404

2. Братко, И. Алгоритмы искусственного интеллекта на языке PROLOG / И. Братко. – М. : Вильямс, 2004. – 640 с.

Е. В. Трусов

Пензенский государственный университет, Пенза, Россия

QT QUICK CONTROLS

Рассказано о библиотеке qt quick controls. Описаны ее возможности, положительные стороны и недостатки.

Qt известна благодаря своей кроссплатформенности позволяющая делать гибкие GUI приложения, с удобным интерфейсом. В него встроен редактор форм с большим набором встроенных компонентов. Для стандартного приложения их вполне достаточно, но порой разработчикам хочется добавить свой собственный компонент, с индивидуальной графикой и логикой. Qt quick controls с qml позволяет это сделать.

Еще в qt 4.7 появилась возможность создавать пользовательский интерфейс (qt quick controls). На протяжении долгого времени разработчики занимались исправлением недочетов и доработками. Но все время не хватало некоторых возможностей: правильного масштабирования на High DPI экранах; компонентов, специфичных для мобильных устройств и стилизации уже существующих компонентов.

Но первые две проблемы решались разработчиками при помощи умножения размеров на коэффициент и написанием собственных компонентов соответственно. Позже в qt quick controls появились готовые пользовательские компоненты в открытом доступе, но это не сильно спасло ситуацию.

Начиная с версии qt 5.7 quick controls потерпел многие положительные изменения и стал называться quick controls 2. В новой версии многое было переписано практически с нуля, были решены проблемы первой версии. Что касается стилизации, то она потерпела большие переработки: отлов событий вынесли в c++, избавились от некоторых ненужных классов, потребляющих много ресурсов устройства (для фона и надписи, теперь вместо них можно использовать что угодно), время запуска приложений сократилось в разы (в 1-й версии для создания слайдера требовалось ~400 мс, во 2й меньше 20).

Qt quick controls может с успехом использоваться в играх со сложным UI. С его помощью создание собственной библиотеки для отображения и инструментов для удобного редактирования может упроститься в разы. Можно и по старинке использовать такие библиотеки как scaleform, coherent UI, awesomium, но у них всех есть очень большой недостаток: огромные тормоза, особенно это заметно на мобильных устройствах.

Что касается причины из-за чего в играх не используют qt widgets – они конфликтуют с такими библиотеками как openGL или directX, Каждый из них хочет быть хозяином главной нити, а потому их совместное использование невозможно.

Но qt quick отлично справится с задачей отрисовки: все его компоненты рисуются ускоренно, а возможность динамического изменения UI программно добавляет скорости, так как не требуется полностью перерисовывать всю модель. Это отлично подходит для динамического изменения внешнего вида игры, быстрой настройки и т.п.

Принципы работы интерфейса, написанного на qml, в следующем: вначале вы описываете различные объекты, располагаете их относительно друг друга (в qml такой же динамически изменяемый интерфейс, как в WPF, а это значит, что координаты фиксировать не нужно). Дальнейшую отрисовку и отображение qt берет на себя.

Помимо этого, qml тесно интегрирован с JavaScript, в результате в компоненты очень удобно встраивать логику: привязки событий, таймеры и просто вспомогательные функции. Все описания можно делать как внутри qml-файлов, так и выносить в отдельные js-файлы.

Что касается конструктора форм, то он не совершенен, т.к. не позволяет редактировать все свойства объекта. Но эти проблемы можно преодолеть, описывая недостающие свойства при помощи текстового редактора.

Так же у qt есть существенный минус: если вы думаете, что после того, как вы написали свой код, нарисовали красивый UI и упаковали ресурсы, у вас все готово к релизу – то это не совсем так.

Стоит напомнить, что qt – это огромное количество dll и qml-модулей. Для того, чтобы иметь возможность переносить программу с пк на пк, все задействованные в программе модули нужно переносить на устройство (именно при помощи их обеспечивается кроссплатформенность). Но чтобы их таскать, сначала нужно найти, а они находятся в разных местах огромной установочной директории. При использовании qt creator, при сборке release версии необходимые модули будут скопированы в папку с приложением, а при использовании другой IDE, придется самому искать все нужные dll и модули.

При разрабатываете игры под pc, или создания крупного мобильного приложения 25–40 Мб библиотек не будут проблемой. Если же вы создаете игру под android, с рекомендованными 50 Мб на приложение, то стоит подумать, нужно ли связываться с данной библиотекой.

Если вам для проекта нужна готовая UI библиотека, а написать свою нет времени, то qtQuick/qml станет хорошим выбором, так как выигрывает по производительности у конкурентов.

При интеграции в проект qt могут возникнуть проблемы: придется изменить инициализацию и логику основного цикла. При создании нового проекта это можно пережить, а быстро сменить UI с одного на qtQuick/qml вряд ли получится.

К qtQuick/qml прилагается подробная документация расписаны большинство функции, есть пояснения с примерами, иногда этого недостаточно. В подобных случаях, есть возможность заглянуть в исходный код (он полностью открыт).

Если сравнивать qml с чем-то похожим, например, xaml, то можно выделить явные преимущества: он позволяет создавать кроссплатформенные программы и очень тесно завязан с JavaScript. Создание новых компонентов происходит очень просто, всю логику можно поместить в него и основной код будет не захламлен. В итоге, я хочу сказать, что плюсы данная библиотека очень хороша, хоть и не сильно известна, она имеет сильные стороны, перевешивающие ее минусы.

Библиографический список

1. Официальная документация по библиотеке qt quick controls. – URL: http://doc.qt.io/qt-5/qtquickcontrols-index.html

2. Шлее, М. Qt 4.8 Профессиональное программирование на C++ / М. Шлее. – СПб : БХВ-Петербург, 2013.

Д. А. Родионов

Пензенский государственный университет, Пенза, Россия

АВТОМАТИЗИРОВАННАЯ СИСТЕМА КОНТРОЛЯ РАБОТОСПОСОБНОСТИ НА БАЗЕ СБИС

Заводы и предприятия, выпускающие радиодетали и микросхемы, после изготовления, но до отправки их на склад, подвергают их контролю на работоспособность. Однако радиодетали и микросхемы, пройдя отдел технического контроля (ОТК) на заводе, имеют некоторый процент брака в процессе транспортировок, монтажа или эксплуатации. Это влечет дополнительные затраты рабочего времени, средств для их выявления и замены (причем большую часть времени занимает именно выявление неисправных деталей).

Особенно важна полная работоспособность комплектующих деталей при сборке ответственных узлов управляющих систем, когда неисправность какого-либо одного элемен-

та микросхемы может повлечь за собой выход из строя других элементов, узлов, а возможно, и всего комплекса в целом.

Объединение аналоговых, цифровых и радиочастотных цепей в одной интегральной микросхеме и увеличение плотности размещения элементов микросхемы приводит к росту числа выводов в новых поколениях микросхем. В результате усложняется процедура тестирования таких микросхем и увеличивается время их выполнения. Это влияет как на стоимость готовых изделий, так и на длительность производственного цикла. Таким образом, производители интегральных микросхем (ИМС) постоянно решают задачу повышения производительности тестирующей аппаратуры, при этом требования, которые предъявляются к ней, порой оказываются на грани технических возможностей. Например, при увеличении числа одновременно тестируемых элементов увеличивается число измерительных и питающих блоков. В то же время стандартные шины, посредством которых обеспечивается синхронизация и необходимый порядок проведения измерений, ограничивают информационный обмен как по скорости, так и по количеству участников.

Здесь хорошую помощь могло бы оказать устройство, которое позволяет очень быстро проверять работоспособность микросхем. Неплохо было бы иметь возможность проверки как новых (подготовленных для монтажа), так и уже демонтированных из платы ИМС. Проверять микросхему очень удобно, когда на плате предусмотрен разъем. Это позволяет проводить достаточно быструю проверку деталей, уменьшить время производства аппаратуры. Это позволит свести к минимуму риск ее выхода из строя, так как в этом случае полностью исключается ее нагрев и различные механические повреждения при монтаже/демонтаже.

В настоящее время на заводах очень большую популярность получили персональные IBM-совместимые компьютеры. Так как задача тестирования ИМС требует наличия высокоинтеллектуального устройства для выполнения алгоритмов тестирования, целесообразно проектировать именно приставку к компьютеру, которая будет подключаться через внешний порт, а не отдельное самостоятельное устройство. Это обусловлено наличием в стандартном комплекте IBM-совместимого компьютера многих компонент, которые необходимы при решении таких задач (микропроцессора, составляющего основу компьютера; жесткого диска, предназначенного для хранения информации; внешних портов ввода-вывода – последовательных COM1, COM2 и параллельного LPT и порта USB; клавиатуры и дисплея – для ввода и вывода информации соответственно).

Автором данной работы разработана относительно недорогая автоматизированная система контроля (АСК), подключаемая к IBM-совместимому компьютеру, предназначенная для тестирования электронных модулей на базе СБИС. АСК должна формировать управляющие сигналы, предназначенные для тестирования электронных модулей.

Проверка производится следующим образом:

К компьютеру посредством кабеля подключается АСК. В разъем, выведенного на его корпус, вставляется испытуемый модуль. На компьютере запускается управляющая программа. Она формирует управляющие сигналы в порт, которые в свою очередь поступают на входы модуля. Далее программа считывает сигналы с выходов проверяемого модуля, анализирует считанные данные и выводит на дисплей результат тестирования.

Устройство предназначено для:

1. Формирования управляющих сигналов, предназначенных для тестирования электронных модулей на базе СБИС, согласно электрической принципиальной схемы на проверяемый модуль;

В современных сложных цифровых печатных платах (ПП), в которых содержаться ПЛИС, имеется JTAG-интерфейс, который подключается к микросхемам.

Идея тестирования микросхем с помощью JTAG-интерфейса – это внутрисистемное тестирование или, как его еще называют, граничное сканирование (Boundary Scan). Суть метода заключается в тестировании ИМС через специальный небольшой разъем при помощи 4-проводного интерфейса, закрепленного стандартом IEEE 1149.1 еще в 1990 году. Этот интерфейс также часто называют JTAG-интерфейсом по названию группы специалистов, которые занимались его разработкой в период с 1985 по 1990 годы (Joint Test Access

Group). Имея в составе устройства одну или несколько микросхем, поддерживающих стандарт IEEE 1149.1, можно тестировать не только цепи, которые связанны с этими микросхемами, но и остальные элементы этой микросхемы, включая память, логику, резисторы и внешние разъемы. При переходе в режим граничного сканирования JTAG-компоненты отключают свою основную логику и переходят в режим тестирования, позволяя внешнему оборудованию управлять своими выводами и тестировать цепи. Инструменты, которые предназначены для тестирования при помощи JTAG-интерфейса, позволяют также провести внутрисистемное программирование Flash-памяти и ПЛИС. Огромное количество микросхем, используемых в современных разработках, поддерживают стандарт IEEE 1149.1 – это требование современной электронной индустрии.

Следует отметить, что JTAG-сканирование можно осуществлять и на системном уровне. Несколько устройств, которые объединены в одну плату, могут иметь общую JTAG-шину граничного сканирования и тестироваться в составе сложного изделия через один единственный разъем. Такое решение часто используется при производстве телекоммуникационного оборудования. Многие предприятия, в особенности оборонного и аэрокосмического приборостроения, практикуют проведение испытаний изделий в экстремальных условиях (критические температуры, повышенная влажность и т.п.). В этом случае граничное сканирование также является незаменимым помощником, так как позволяет проводить тестирование и определять «слабые» места на плате непосредственно в камере для испытаний.

Порт тестирования (TAP–TestAccessPort) представляет собой четыре или пять выделенных линий микросхемы: TCK, TMS, TDI, TDO и TRST(опционально).

Функциональное назначение этих линий:

- **TDI**(«вход тестовых данных») – вход последовательных данных периферийного сканирования. Команды и данные поступают в микросхему с этого вывода по переднему фронту сигнала линии TCK;
- **TDO**(«выход тестовых данных») – выход последовательных данных. Команды и данные выводятся из микросхемы с этого вывода по заднему фронту сигнала TCK;
- **TCK**(«тестовое тактирование») – тактирует работу встроенного автомата управления периферийным сканированием. Максимальная частота сканирования периферийных ячеек зависит от используемой аппаратной части и на данный момент ограничена 25…40 МГц;
- **TMS**(«выбор режима тестирования») – обеспечивает переход микросхемы в/из режима тестирования и переключение между различными режимами тестирования.
- **TRST**(«тестовый сброс») – данный сигнал добавляется в определенных случаях для инициализации порта тестирования, что необязательно, так как инициализация возможна путем подачи определенной последовательности сигналов на вход линии TMS.

Состояния автомата определяются сигналами линий TDI и TMS порта тестирования. Определенное сочетание сигналов TMS и TCK обеспечивает ввод команды для автомата и ее исполнение.

При тестировании сложных цифровых устройств граничное сканирование является наиболее оптимальным методом, учитывая относительно невысокую стоимость оборудования, низкие трудозатраты, а также то, что возможность тестирования уже заложена во многие цифровые микросхемы.

2. Автономной проверки электронных модулей (без ПК);

Для выполнения в большом количестве простых однотипных операций, не требующих наличие ПК, таких как тестирование кабелей, программирование ИМС, предусмотрен режим автономной проверки.

3. Входного контроля ИМС.

Входной контроль компонентов является неотъемлемой частью обеспечения качества продукции в производстве электроники. С повышением уровня сложности производимых изделий, повышается уровень ответственности компонента в изделии.

Проведение входного контроля необходимо для любых типов компонентов, от резисторов и до ИМС. Очень важно на этапе входного контроля оценить работоспособность ИМС.

Обычно, производители электроники на входном контроле ограничиваются визуальным осмотром и инструментальным контролем геометрии микросхем.

Однако, при отсутствии внешних повреждений корпуса и соответствии чертежам, могут наблюдаться дефекты ИМС, выявляемые лишь тестированием компонента. При проведении тестирования ИМС, ее серия и тип известны, и микросхема считается исправной, при условии соответствия всех контролируемых входных и выходных сигналов для данной ИМС нормативным требованиям технических условий.

АСК реализовывает следующие функции:

1) независимое управление цифровыми каналами (ЦК);
2) управление АСК с ПК;
3) автономная работа системы.

АСК удовлетворяет следующим требованиям:

1) напряжение питания – 5В;
2) количество независимых ЦК – 160;
3) управление по интерфейсу USB.

В результате исследований были выявлены следующие аналоги АСК:

1) PCI-1753 является 96-канальной платой цифрового ввода/вывода с шиной PCI. Для организации 96 каналов цифрового ввода/вывода каналы разделены на двенадцать 8-битных портов. Каждый порт может быть запрограммирован либо как порт ввода либо как порт вывода. Цена PCI-1753 составляет 21 700 руб. Недостатком этого устройства является невозможность одновременной побитовой настройки портов на прием/передачу сигналов. Цена разрабатываемого устройства при этом составляет 21 000 руб.;

2) USB-4750 – это 32-канальный модуль цифрового ввода/вывода с USB интерфейсом. Настройка работы данного модуля не вызывает затруднений, так как каждый его канал ввода/вывода соотносится с битом в порте ввода/вывода. Цена устройства составляет 21 500 руб. Недостатком этого устройства является малое количество ЦК;

3) NI 6581 представляет собой 54-канальный модуль цифрового ввода/вывода с различными вариантами напряжения – 1,8 В, 2,5 В, 3,3 В. Цена устройства составляет 82 000 руб. и это является его недостатком. Также для работы NI 6581 необходимо дополнительное дорогостоящее оборудование.

В табл. 1 отображены все положительные и отрицательные качества рассмотренных ранее аналогов по отношению к АСК.

Таблица 1

Сравнение аналогов разрабатываемого устройства

Наименование	Цена	Кол-во ЦК	Простота организации структуры
PCI-1753	+	–	–
USB-4750	+	–	–
NI 6581	–	–	–
АСК	+	+	+

Из таблицы видно, что АСК обладает большим преимуществом относительно его аналогов. Это достигается за счет использования другой, более простой элементной базы, реализации управления АСК при помощи USB-интерфейса и формирования управляющих сигналов различной формы.

Библиографический список

1. Ключев, А. О. Использование интерфейса JTAG для отладки встраиваемых систем / А. О. Ключев, Т. А. Коровьякова, А. Е. Платунов // Изввестия вузов. Приборостроение. – 2014. – 341с.

2. Алексеенко, А. Г. Проектирование радиоэлектронной аппаратуры на микропроцессорах: Программирование, типовые решения, методы отладки / А. Г. Алексеенко, А. А. Галицын, А. Д. Иванников. – М. : Радио и связь, 2014. – 272 с.
3. URL: http://kit-e.ru/articles/measure/2005_6_230.php (дата обращения: 14.09.2017).
4. URL: http://www.studfiles.ru/preview/1544154/ (дата обращения: 13.09.2017).

С. В. Шибанов, И. Ю. Балашова, П. П. Макарычев

Пензенский государственный университет, Пенза, Россия

ПЕРСПЕКТИВЫ И ПРОБЛЕМЫ ПРИМЕНЕНИЯ БЛОКЧЕЙНОВ ДЛЯ ПОСТРОЕНИЯ РАСПРЕДЕЛЕННЫХ КОРПОРАТИВНЫХ СИСТЕМ

Рассмотрены проекты на основе блокчейнов. Выделены проблемы распределенных блокчейн-систем и предложены подходы к их решению.

Используемые в настоящее время в финансовых и государственных учреждениях информационные технологии морально устарели. Ориентация на максимальное сближение с клиентом требует передачи права на доступ к данным третьим лицам и приложениям при существующем по умолчанию недоверии между участниками операций. Ограниченные возможности интеграции данных из различных систем не обеспечивают доступность множества услуг большому количеству различных клиентов при сохранении (а в идеале, и повышении) уровня надежности и безопасности систем. Сложность внутренней организации обуславливает высокие расходы на эксплуатацию и сопровождение. Вышеперечисленные проблемы вынуждают в процессе развития корпоративных информационных систем ориентироваться, прежде всего, на использование децентрализованной распределенной архитектуры.

Впервые о технологии блокчейн заговорили в контексте криптовалюты биткойн, которая в настоящее время занимает первое место в рейтинге ведущих блокчейн-систем. Для биткойна блокчейн реализован в виде распределенного аналога гроссбуха (ledger), в котором регистрируются все новые транзакции с биткойном, после чего их изменение или отмена становятся невозможными. После многолетнего успешного функционирования в качестве программной основы криптовалюты биткойн, технология блокчейн заинтересовала разработчиков как возможное эффективное решение для реализации многих не связанных с виртуальными деньгами проектов. Многие крупные финансовые организации и международные корпорации, такие, как IBM, Microsoft, Goldman Sachs, Barclysm BNYMellon и другие, начали инвестировать в исследования и разработку блокчейн-технологии. В первую очередь данная технология привлекла внимание банков и других финансовых организаций, поскольку она повышает защищенность от кибератак, гарантирует неизменность данных, снижает операционные расходы. В январе 2017 г. семь крупнейших европейских банков (Deutsche Bank, HSBC, KBC, Natixis, Rabobank, Societe Generale и UniCredit) объединились в создании общей финансовой платформы для трансграничной торговли на основе блокчейн-технологии. Qiwi публично объявила о регистрации товарного знака «битрубль». К интересным финансовым проектам можно отнести «Ripple», совмещающий в себе распределенную платежную систему и валютную биржу.

Предпринимаются попытки создания на основе блокчейна систем электронного документооборота в пределах организаций или территориальных образований. Сбербанк анонсировал запуск «Digital Ecosystem» – альтернативной системы электронного документооборота на основе технологий блокчейн, ключевым отличием которой является отсутствие единого провайдера и открытость для всех участников. Использовать «Digital

Ecosystem» на первом этапе помимо крупнейшего российского банка планируют Федеральная антимонопольная служба (ФАС), «Аэрофлот», ОАК, «Русский уголь» и «ФортеИнвест». Корпорации Mahindra Group и IBM объявили о создании блокчейн-решения для финансовой логистики, позволяющего повысить безопасность, прозрачность и качество операционных процессов в цепочке поставок. Имеются опробованные блокчейн-системы, эксплуатирующие такое преимущество блокчейна, как возможность проведения операций обмена без посредников. Это всевозможные децентрализованные биржи (NXTAsset Exchange, Bitshares, OpenLedger), краудфандинговые площадки, системы p2p кредитования.

Другие направления развития применения блокчейна – это единые учетные реестры, отчетность, системы голосования и опросы, нотариальные документы, оформление товаров (например, на уровне таможенных документов), государственное финансирование и многосторонние услуги. Известно, что Национальный расчетный депозитарий РФ завершил разработку и успешно протестировал прототип системы электронного голосования для собраний владельцев облигаций «Электронное каскадное голосование» (E-proxy-voting) на основе технологии блокчейн. В конце января 2017 г. стало известно о создании нового блокчейн-консорциума, в задачи которого входит повышение безопасности и улучшение использования Интернета вещей. В организацию вошли Cisco, Bosch, Foxconn, Gemalto, банк BNY Mellon, а также блокчейн-стартапы Consensus Systems, BitSE и Chronicled. Совместными усилиями компании будут разрабатывать на основе технологии блокчейн протокол безопасного обмена информацией между всевозможными IoT-устройствами (стиральные машины, транспортированные контейнеры, датчики и т.п.)

Важный шаг в развитии технологии блокчейн был сделан проектом «Ethereum», предложившим универсальную блокчейн-платформу для создания приложений на основе смарт-контрактов – программных модулей, выполняющих алгоритм сделки с произвольными ифровыми активами. Главными преимуществами смарт-контрактов по сравнению с обычными контрактами являются безопасность, отсутствие необходимости в посредниках, нижение транзакционных издержек и, конечно же, автоматичность выполнения. На платформе «Ethereum» реализован сервис децентрализованных автономных организаций ДАО). ДАО представляют собой систему смарт-контрактов, полностью или частично обеспечивающую работу коммерческих и некоммерческих организаций. Возможности Ethereum могут быть применяются для разработки, децентрализации, и защиты решений в самых разных областях человеческой деятельности: голосования, доменные имена, финансовые биржи, краудфандинг, управление компаниями, большинство разновидностей контрактов и договоров, интеллектуальные права и даже умная собственность, в случае интеграции с аппаратными решениями. Microsoft совместно с компанией ConsenSys предоставила в Azure средства разработки для платформы «Ethereum» по модели «Ethereum Blockchain as a Sevice» (EBaaS). EBaaS позволяет клиентам быстро развернуть и протестировать свои решения на базе технологии блокчейна, освобождая их от каких-либо капитальных затрат на этапе апробации новой технологии. Готовое решение может развертываться в любых публичных облаках, в частных облаках, на виртуальных и физических машинах. Корпорация IBM прилагает усилия к развитию корпоративного применения технологии блокчейн, в том числе предлагая блокчейн в качестве услуги разработчикам. Новый набор связанных с блокчейн услуг в облаке IBM Cloud помогает разработчикам создавать и управлять блокчейн-сетями. Разработчики могут генерировать цифровые активы и сопутствующую бизнес-логику, способствующую более безопасной и конфиденциальной передаче активов между участниками тестовой сети блокчейн с защищенными правами доступа.

Опубликованные государственные и негосударственные аналитические исследования констатируют, что использование технологии блокчейн при разработке распределенных корпоративных информационных систем снизит издержки финансовых операций, одновременно повысив их защищенность от мошенничества, а также даст неоспоримые преимущества при организации контроля над всевозможными активами и обеспечении конфиденциальности данных. Однако необходимо отметить ряд основных технических

проблем, с которыми сталкиваются блокчейн-системы: плохая масштабируемость (рост объема данных в каждом узле), тенденция к централизации. проблема доверия к данным.

В данный момент в разработке находится проект, целью которого является создание платформы для построения высокоэффективных и безопасных децентрализованных распределенных информационных систем на основе технологии блокчейн. В рамках данного проекта проблему больших объемов хранения данных на всех узлах распределенной сети и, как следствия – плохой масштабируемости, предлагается решать за счет лимитирования длины блокчейна и введения промежуточного узла доверенного подтверждения хронологически отдаленных от текущего момента транзакций. Данные узлы являются гарантами достоверности предыдущих транзакций, при этом на узлах распределенной сети хранятся блокчейны только с транзакциями за лимитированный промежуток времени относительно текущего момента. Блокчейн при этом строится от фиксированного слепка цепочки, хранящегося на узлах доверенного подтверждения. Достоверность слепка подтверждается пересчетом на всех узлах сети и голосованием, после чего он фиксируется и не подлежит изменениям. Таким образом, откат и разрешение коллизий для удаленных хронологически от текущего момента транзакций становится невозможным, но при этом снижается вычислительная нагрузка при выполнении достоверности текущих транзакций.

Проблему стремления к централизации системы как следствия растущей ресурсоемкости предлагается решать путем применения процедуры подтверждения достоверности данных за счет динамического перераспределения весов узлов с использованием нейросетевых алгоритмов.

Любые данные, размещаемые в блокчейн-системах, обладают не большей достоверностью, чем исходная оффлайн-информация. То есть использование технологии блокчейн обеспечивает лишь сохранность данных, защиту их от модификации после их ввода в систему и уверенность в однозначности транзакций с ними внутри блокчейна, но не гарантию их полной достоверности. Такого же рода ограничения существуют и при работе со смарт-контрактами – они могут работать только при условии, что исполняемый код имеет прямой доступ к написанным в них объектам, то есть могут оперировать только данными, сохраненными в реализующей смарт-контракты блокчейн-платформе. Поэтому контракты, функционирующие только внутри замкнутой информационной среды конкретного блокчейна, не могут широко применяться государственными, финансовыми и бизнес-институтами. Для обеспечения максимального доверия к вводимым в систему внешним данным предлагается использовать оффлайн-протоколов на основе математически доказуемых формализмов с использованием логики предикатов.

Опыт коммерческого и общественного использования технологии блокчейн в разных странах показывает ее явные преимущества (простота, прозрачность, дешевизна, надежность, скорость) по сравнению с традиционными процедурами. Технологии блокчейн лучше всего применяются в тех сферах, где требуется хранение, постоянная обработка и обновление больших объемов защищенных данных.

Библиографический список

1. Мазонка, О. Хэш Цепочки Технологии Hasq / О. Мазонка, В. Попов. – Hasq Technology Pty Ltd, Австралия, 2014.

2. BigchainDB: A Scalable Blockchain Database / T. McConaghy, R. Marques, A. Muller, D. De Jonghe, T. Troy McConaghy, G. McMullen, R. Henderson, S. Bellemare, A. Granzotto. – ascribe GmbH, Berlin, Germany, 2016.

3. Distributed Ledger Technology: beyond block chain. A report by the UK Government Chief Scientific Adviser. Available at:. – URL: https://www.gov.uk/government/uploads/system/uploads/ attachment_data/file/492972/gs-16-1-distributed-ledger-technology.pdf (accessed: 4 April 2017).

4. Unlocking economic advantage with blockchain: a guide for asset managers. Available at. – URL: http://www.oliverwyman.com/our-expertise/insights/2016/jul/unlocking-economic-advantage-with-blockchain.html (accessed: 7 April 2017).

С. В. Шибанов, А. В. Тарасеев, Е. С. Кикта

Пензенский государственный университет, Пенза, Россия

КЛАССИФИКАЦИЯ ПРАВИЛ В АКТИВНЫХ БАЗАХ ДАННЫХ

Рассматриваются активные правила как механизм реагирования на внутренние и внешние события в активных базах данных. Исследуются основные свойства активных правил, приводится классификация правил по наиболее важным критериям.

Активные базы данных (АБД) поддерживают механизмы, позволяющие им автоматически отвечать на события, которые происходят как внутри, так и снаружи системы. Для реагирования на события в АБД используются так называемые активные правила, которые содержат три основных компонента – событие, условие и действие. Событие (event) описывает то, что должно произойти, и на что правило должно быть способно реагировать. Условие (condition) проверяет контекст, при котором событие произошло. Действие (action) описывает процедуру, которая должна быть выполнена правилом, если соответствующее событие произошло и условие оказалось истинным [1].

На сегодняшний день существует большое количество различных моделей активных правил. При решении конкретной задачи, которая подразумевает применение какой-либо модели правил, перед разработчиком встает вопрос: «Какая именно концепция активных правил будет наиболее эффективна в данном случае?». Чтобы облегчить столь сложный выбор, необходимо классифицировать активные правила, предварительно определив критерии классификации.

Прежде всего, необходимо отметить, что в АБД описываются и обрабатываются два основных типа событий:

- примитивное (простое, элементарное) событие, которое порождается одиночным низкоуровневым действием;
- сложное (составное, композитное) событие, которое порождается комбинацией примитивных или сложных событий, используя ряд операторов, которые составляют алгебру событий [2].

Правила, реагирующие на примитивные события, можно классифицировать с учетом источника возникновения события на следующие виды:

- структурные, которые порождаются операцией над некоторой частью структуры (например, вставить кортеж, изменить атрибут, получить доступ к кортежу);
- поведенческие, которые порождаются выполнением некой операции, описанной пользователем;
- транзакционные, которые порождаются командами транзакций (например, abort, commit, begin-transaction);
- абстрактные, которые позволяют сигнализировать о возникновении события явно, за счет механизма программирования (например, в ответ на некоторую информацию, введенную пользователем);
- срабатывающие на исключения, которые порождаются в результате некоторого возникшего исключения (например, была сделана попытка получить доступ к данным без соответствующей авторизации);
- временные, которые возникают в некоторый момент времени. Часто их подразделяют на абсолютные (например, 13 Ноября 1998 в 15:00), относительные (например, 10 дней после того, как акции проданы) и периодические (например, в первый день каждого месяца);
- внешние, которые порождаются чем-то извне (например, измеряемая температура выше 30 градусов).

По масштабу события (Event Granularity) правила могут быть определены:

- для всего набора элементов (например, все экземпляры класса);
- для заданных поднаборов (например, все штатные сотрудники, кроме профессоров);
- для некоторых элементов набора (например, чтобы предотвратить неавторизованный доступ к некоторым экземплярам, или чтобы сделать возможным обновление тех объектов виджета, которые сейчас находятся на экране) [3].

Набор операция, с помощью которых описываются сложные события, как комбинации примитивных или других сложных событий называется алгеброй событий. В [2] предлагается ряд операторов, на основании которых можно классифицировать сложные события:

- использующие оператор «дизъюнкция» – E1 или E2 возникает, когда возникает либо E1, либо E2;
- использующие оператор «конъюнкция» – E1 и E2 возникает, когда возникают оба события E1 и E2 в любом порядке;
- использующие оператор «последовательность» – seq(El,E2) возникает, когда E1 появляется раньше E2;
- использующие оператор «закрытие» – closure E in Int возникает только один раз, когда появляется E, и блокирует последующие появления E в течение интервала Int;
- использующие оператор «история» – times(n,E) in Int возникает, когда E появляется n раз в течение временного интервала Int;
- использующие оператор «не» – not El in Int обнаруживается, если событие El не появлялось в течении интервала Int;
- использующие связки нескольких операторов [1].

Существуют два основных режима связывания правил компонентов правила «событие-условие» и «условие-действие». Первый определяет, когда условие вычисляется относительно события, инициировавшего правило. Режим связывания условие-действие показывает, когда действие должно быть выполнено относительно вычисления условия. На основании данных режимов выделяют следующие виды правил:

- имеющие непосредственный (immediate) режим связывания, в случае которого условие (действие) вычисляется (выполняется) сразу после события (условия);
- имеющие отложенный (deferred) режим связывания, в случае которого условие (действие) вычисляется (выполняется) в той же транзакции, что и событие (условие) правила, но не обязательно в первую очередь. Однако, в некоторых моделях описывается режим связывания, осуществляемый пользователем, когда условие (действие) вычисляется (выполняется) в описанное пользователем время после того, как событие (условие) просигналит (вычислится). Похожий эффект также поддержан в системе Starburst, где пользователи могут вызвать обработку правил внутри транзакции специальными командами: команды «обработать правила», «обработать список правил S» и «обработать правило R» для обработки всех правил, заданного набора или одного правила, соответственно;
- имеющие отдельный (detached) режим связывания, в случае которого условие (действие) вычисляется (выполняется) в другой транзакции, относительно события (условия). Выполнение действия может зависеть или не зависеть от подтверждения транзакции, в которой было событие или вычислялось условие [4].

Когда несколько правил вызываются в одновременно в качестве реакции на некоторое событие, возникает проблема выбора следующего правила для запуска. На основании этого выделяют:

- динамические правила, в которых назначение приоритетов правилам основано либо на давности обновления (то есть на времени появления события), либо на сложности условия;
- статичные правила, в которых приоритеты часто назначаются либо системой (т.е. на основе времени создания правила), либо пользователем как атрибут правила. Правила могут быть расположены по порядку, используя числовую схему, в которой каждому правилу дается абсолютное значение, являющееся приоритетом, или указывая относительные приоритеты правил явным определением того, что данное правило должно быть запущено перед другим, когда оба они вызываются одновременно.

При исполнении сложных правил в одно и то же время не менее важен выбор числа правил, которые нужно запустить. Поэтому их можно разделить на те, в которых:

- запуск всех экземпляров правил осуществляется последовательно;
- запуск всех экземпляров правил происходит параллельно;
- есть возможность запустить все экземпляры конкретного правила перед любыми другими;
- есть возможность запустить только один или несколько экземпляров правил [5, 6].

Таким образом, на основании данной классификации, среди множества существующих типов активных правил каждый разработчик сможет найти такую модель, которая наиболее оптимально подойдет при решении конкретной задачи.

Библиографический список

1. Paton, N. Active Database Systems / N. Paton, O. Diaz // ACM Computing Surveys. – March 1999.

2. Dayal, U. Active Database Systems / U. Dayal, E. Hanson, J. Widom // Interoperability, and Beyond. – Addison-Wesley, Reading, Massachusetts, Sep. 1994.

3. Шибанов, С. В. Графовые модели для анализа взаимодействий правил в активных базах данных / С. В. Шибанов, А. Б. Зудов // Модели, системы, сети в экономике, природе и обществе. – Пенза, 2016. – № 1 (17). – С. 344–354.

4. Шибанов, С. В. Формальное представление правил в активных базах данных как последовательных взаимодействующих процессов / С. В. Шибанов, А. Б. Зудов // Модели, системы, сети в экономике, природе и обществе. – 2016. – № 1 (17). – С. 335–343.

5. Шибанов, С. В. Формализация ECA и SECA-моделей активных правил / С. В. Шибанов, Э. В. Лысенко, А. А. Скоробогатько // Прикладная математика, управление и информатика : сб. тр. Междунар. молодеж. конф. : в 2 т. – Белгород : ИД «Белгород», 2012. – Т. 1. – С. 319–322.

6. Шибанов, С. В. Платформа для разработки приложений активных баз данных / С. В. Шибанов, А. А. Скоробогатько, А. Б. Зудов, П. В. Вишняков // Научный сервис в сети Интернет: поиск новых решений : тр. Междунар. суперкомпьютерной конф. – М. : Изд-во МГУ, 2012. – С. 591–599.

А. Р. Бородулин

Пензенский государственный университет, Пенза, Россия

ЯЗЫК COLAMO И КОМПЛЕКС СРЕДСТВ ПРОГРАММИРОВАНИЯ МНОГОКРИСТАЛЬНЫХ РЕКОНФИГУРИРУЕМЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

Обсуждаются возможности современных инструментальных средств для проектирования аппаратных средств, ускоряющих работу вычислительных систем при реализации вычислительно трудоемких алгоритмов.

Одним из перспективных способов достижения высокой реальной производительности вычислительной системы является адаптация ее архитектуры под структуру решаемой задачи и создание вычислительного устройства, одинаково эффективно реализующего как структурные, так и процедурные фрагменты вычислений. Поэтому отечественные и зарубежные производители вычислительной техники все чаще включают в состав разрабатываемых вычислительных систем программируемые логические интегральные схемы (ПЛИС) для ускорения работы при реализации вычислительно трудоемких алгоритмов. Создаются как отдельные ускорители с одним-двумя кристаллами ПЛИС, так и целые вычислительные комплексы. Некоторые выпускают ряд ускорителей и несущих плат с не-

большим числом (до четырех) кристаллов ПЛИС, которые используются в создании серверов и гибридных кластерных систем фирмами HP и IBM.

В настоящее время существует множество различных систем проектирования, позволяющих создавать структурные решения прикладных задач на ПЛИС. Наиболее популярными как в качестве отдельного средства разработки, так и в составе комплексов, являются программы- синтезаторы, предоставляемые фирмами-изготовителями ПЛИС: ISE и Vivado фирмы Xilinx, Quartus II фирмы Altera и Actel Libero IDE фирмы Actel. Данные средства, по- мимо непосредственно среды проектирования цифровых устройств, включают в себя ряд вспомогательных утилит: анализаторы временных характеристик, редакторы размещения, модули программирования ПЛИС, системы моделирования цифровых устройств и др. Благодаря широкому инструментарию эти системы проектирования обеспечивают полный цикл разработки цифровых устройств в пределах одного кристалла ПЛИС: создание исходного описания проекта, синтез, моделирование, размещение, трассировку, конфигурирование кристалла.

В связи с постоянным ростом емкости кристаллов ПЛИС проектирование решений прикладных задач в кристаллах ПЛИС с помощью языков описания аппаратуры (VHDL, AHDL, Verilog и др.), а также разработка цифровых устройств в графических редакторах становится все более трудоемкой. Поэтому в настоящее время ведущие производители ПЛИС и реконфигурируемых вычислителей ориентируются на языки высокого уровня. Так, в новой среде разработки Vivado фирмы Xilinx добавлен новый инструмент проектирования Vivado HLS, основанный на языке высокого уровня, а компания Altera для своих ПЛИС предлагает инструментарий разработки Altera SDK (http://www.altera.com/literature/lit-opencl-sdk.jsp, дата обращения 12.12.2014) для нового стандарта параллельного программирования гетерогенных систем OpenCL. В данных решениях используются трансляторы С-подобных языков, генерирующие из программы на аналоге языка программирования высокого уровня С код на языках описания аппаратуры на уровне регистровых передач (RTL-уровень, трансляторы C-to-RTL).

Несмотря на схожесть синтаксисов данных С-подобных языков с самим языком С, подобный подход вовсе не означает, что исходный код на языке С, написанный под персональный компьютер или кластерную вычислительную систему, будет понят трансляторами C-to-RTL. Выбор языка С в качестве основы обусловлен широкой распространенностью данного языка, что существенно упрощает освоение новых инструментариев разработки решений прикладных задач для ПЛИС.

Также при использовании трансляторов C-to-RTL весь код программы либо явно указанные процедуры транслируются в RTL-описания отдельных кристаллов ПЛИС. В подобных системах разработки отсутствует инструментарий, обеспечивающий автоматические разбиения параллельной программы на множество связанных кристаллов ПЛИС.

В Vivado HLS проект разрабатывается в рамках одного кристалла ПЛИС, и если программисту недостаточно аппаратного ресурса кристалла, то он вынужден самостоятельно распределять вычисления между несколькими проектами для каждой ПЛИС и синхронизировать управляющие и информационные потоки между ними.

Стандарт OpenCL используется компанией Nallatech (разработчик реконфигурируемых вычислителей) и подразумевает использование нескольких ПЛИС в одном проекте. Программирование решений в кристаллах ПЛИС в данном случае осуществляется с помощью вызова функций из библиотек инструментария разработки Altera SDK, и в каждом задействованном кристалле ПЛИС выполняются вычисления, описанные отдельным участком кода. Таким образом, программа, написанная в стандарте OpenCL, представляет собой основной код, написанный под традиционные процессоры, и отдельные участки кода, написанные под ПЛИС, задействованные как сопроцессоры. В данном случае задача синхронизации данных также возлагается на самого программиста.

Еще одним известным средством программирования ПЛИС является комплекс, разработанный компанией Mitrionics, состоящий из виртуального процессора Mitrion Virtual Processor (MVP), программирование которого осуществляется на языке высокого уровня Mitrion-C, и библиотеки функций для построения хост-программ MithalAPI, входящих

в пакет разработки 195 Mitrion SDK. Разрабатываемая на языке Mitrion-C программа должна быть полностью реализована на одном виртуальном процессоре MVP, что не позволяет программировать многокристальные РВС, и, следовательно, существенно снижает эффективность использования программного комплекса компании Mitrionics. Для программирования многокристальных РВС со связями между ПЛИС программисту необходимо самому реализовать интерфейс (протокол) обмена данными между ПЛИС и решить вопросы, связанные с синхронизацией потоков данных. В этом случае программа для РВС вырождается в программу для кластера (множество процессоров MVP), реализованного на ПЛИС, что существенно снижает эффективность реали- зации задач на многокристальных РВС.

Альтернативный подход к программированию РВС предлагается в НИИ МВС ЮФУ, занимающемся разработкой многокристальных реконфигурируемых вычислительных систем раз-личных архитектуры и конфигурации уже более 15 лет. Опыт решения различных классов задач в НИИ МВС ЮФУ показал, что для эффективного решения современных трудоемких задач программисту необходимы средства программирова-ния, обеспечивающие:

- программирование на языке высокого уровня;
- поддержку многокристального программирования;
- обеспечение высокой частоты работы ПЛИС;
- высокую плотность заполнения кристалла ПЛИС;
- поддержку конвейерной и макро конвейерной организации вычислений.

В НИИ МВС ЮФУ разработан и широко используется программный комплекс, состоящий из:

- транслятора языка программирования COLAMO, осуществляющего трансляцию исходного кода на COLAMO в информационный граф параллельной прикладной программы;
- синтезатора масштабируемых схемотехнических решений на уровне логических ячеек ПЛИС Fire!Constructor, осуществляющего отображение полученного от транслятора языка программирования COLAMO информационного графа на архитектуру РВС, размещение отображенного решения по кристаллам ПЛИС и автоматическую синхронизацию фрагментов информационного графа в разных кристаллах ПЛИС;
- библиотеки IP-ядер, соответствующих операторам языка COLAMO (функционально-законченных структурно-реализованных аппаратных устройств) для различных предметных областей, и интерфейсов для согласования скорости обработки информации и связи в единую вычислительную структуру;
- средств отладки и программ доступа и мониторинга состояния РВС.

Язык высокого уровня COLAMO предназначен для описания реализации параллельного алгоритма и создания на основе принципов структурно-процедурной организации вычислений [1, 8, 9] в архитектуре РВС специализированной вычислительной структуры, которая предполагает последовательную смену структурно (аппаратно) реализованных фрагментов информационного графа задачи, каждый из которых является вычислительным конвейером потока операндов.

Таким образом, приложение (прикладная задача) для РВС состоит из структурной составляющей, представленной в виде аппаратно реализованных фрагментов вычислений, и процедурной составляющей, представляющей собой единую для всех структурных фрагментов управляющую программу последовательной смены вычислительных структур и организации потоков данных.

Для представления такой организации вычислений в языке используется понятие «кадр». Кадром является программно-неделимый компонент, представляющий собой совокупность операторов, которые реализуются в виде арифметико-логических команд и команд чтения/записи, выполняемых на различных функциональных устройствах, соединенных между собой в соответствии с информационной структурой алгоритма.

В языке COLAMO отсутствуют явные формы описания параллелизма, а распараллеливание достигается с помощью объявления типов доступа к переменным и индексации элементов массивов, что характерно для языков потока данных. Для обращения к данным используется два основных метода доступа: параллельный доступ (задаваемый типом

Vector) и последовательный доступ (задаваемый типом Stream). Степень параллелизма определяется по минимальному значению параметра распараллеливания. Для типа доступа Stream степень параллелизма равна 1, а для типа доступа Vector определяется наименьшим значением векторной составляющей каждого массива, участвующего в вычислениях. Для параллельного типа доступа возможна одновременная обработка всех размерностей массивов, заданных типом Vector, при этом повышается аппаратный ресурс на обработку, но снижается время обработки.

Многомерные массивы данных могут иметь множество измерений, каждое из которых может иметь последовательный или параллельный тип доступа, задаваемый ключевым словом Stream или Vector соответственно. Изменение типа доступа позволяет достаточно просто управлять как степенью распараллеливания вычислений на уровне описания структур данных, так и скоростью обработки, и занимаемым ресурсом, что позволяет программисту описывать различные виды параллелизма в достаточно сжатом виде.

Помимо типа доступа, для переменной в языке COLAMO определены также и типы ее хранения: мемориальный (Mem), регистровый (Reg) и коммутационный (Com).

Мемориальной переменной называется величина, хранящаяся в ячейке распределенной памяти и, следовательно, сохраняющая свое значение до очередного переприсваивания. Для мемориальной переменной возможно одновременное выполнение только одного процесса. Поэтому в семантике языка COLAMO для мемориальной переменной в теле кадра действуют правило однократного присваивания и правило единственной подстановки. Правило однократного присваивания указывает на то, что мемориальная переменная в кадре изменяет свое значение только один раз. Правило единственной подстановки определяет, что переменная в кадре может использоваться только для одного процесса чтения или записи.

Для описания связей между элементами информационного графа задачи в языке COLAMO предназначена коммутационная переменная. Поскольку коммутационная переменная описывает информационные связи, то она не требует никакого вычислительного аппаратного ресурса для своей реализации. Доступ к значению коммутационной переменной после выполнения кадра невозможен. Коммутационная переменная необходима транслятору для указания информационных зависимостей при построении вычислительной структуры задачи. На коммутационную переменную, как и на мемориальную переменную, действует правило однократного присваивания, но для нее не выполняется правило единственной подстановки. Использование коммутационных переменных позволяет легко разветвлять и дублировать потоки данных, но не позволяет реализовать рекурсию.

Для организации рекурсии в языке COLAMO используется регистровая переменная, которая представляет собой регистр на аппаратном уровне и используется для хранения промежуточных данных, полученных в процессе вычислений. Единственным ограничением для регистровой переменной в теле кадра является правило однократного присваивания.

Трансляция программы на языке высокого уровня COLAMO состоит в создании схемотехнической конфигурации вычислительной системы (структурной составляющей) и параллельной программы, управляющей потоками данных (потоковой и процедурной составляющих). Создание структурной составляющей заключается в построении вычислительного графа, соответствующего описанным на COLAMO информационным зависимостям между результатами вычислений.

При этом для каждой используемой в тексте программы операции подставляется специализированный вычислительный блок в зависимости от типа доступа к переменным, типа данных, их разрядности и т.д. Синтезированный информационный граф задачи передается в синтезатор Fire!Constructor для отображения на аппаратный ресурс многокристальной РВС.

Задача автоматического отображения параллельной программы на аппаратный ресурс многокристальной РВС состоит из трех подзадач: разбиения информационного графа на непересекающиеся подграфы, размещения сформированных подграфов в ПЛИС РВС и трассировки внешних связей размещенных подграфов в системе коммутаций РВС.

Результатом работы синтезатора Fire!Constructor являются файлы VHDL-описаний и файлы временных и топографических ограничений (User Constraints Files). Файлы VHDL описывают структурные реализации фрагментов параллельной программы. На основании этих файлов и библиотеки схемотехнических элементов формируются проекты для синтезатора ISE под каждую отдельную ПЛИС. Далее с помощью синтезатора ISE формируются конфигурационные файлы ПЛИС. Полученные конфигурационные файлы загружаются в РВС.

Программа на языке COLAMO разрабатывается в едином проекте и может быть транслирована на любую РВС, описание которой и соответствующие библиотеки присутствуют в комплексе программирования РВС. В отличие от других существующих систем разработки решений прикладных задач на ПЛИС пользователю не требуется в тексте программы указывать, какие фрагменты программы в каких ПЛИС будут выполняться. Синтезатор Fire!Constructor обеспечивает автоматическое разбиение вычислительной структуры программы на языке COLAMO на несколько проектов в синтезаторе Xilinx ISE, при этом обеспечивается синхронизация информационных потоков как внутри кристалла ПЛИС, так и между ними.

Библиографический список

1. Каляев, И. А. Реконфигурируемые мультиконвейерные вычислительные структуры / И. А. Каляев, И. И. Левин, Е. А. Семерников, В. И. Шмойлов ; под общ. ред. И. А. Каляева. – 2-е изд., перераб. и доп. – Ростов н/Д : Изд-во ЮНЦ РАН, 2009. – 344 с.

2. Левин, И. И. Реконфигурируемые вычислительные системы с открытой масштабируемой архитектурой / И. И. Левин // Параллельные вычисления и задачи управления : тр. V Междунар. конф. PACO'2010. – М. : Учреждение Российской академии наук Институт проблем управления им. В. А. Трапезникова РАН, 2010. – С. 83–95.

А. В. Николаев[1], Б. В. Цыпин [2], Д. А. Ярославцева[1]

[1]АО «Научно-исследовательский институт физических измерений», Пенза, Россия
[2] Пензенский государственный университет, Пенза, Россия

ИСПОЛЬЗОВАНИЕ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ ДЛЯ АНАЛИЗА СТРУКТУРНЫХ СХЕМ ИНФОРМАЦИОННО-ИЗМЕРИТЕЛЬНЫХ И УПРАВЛЯЮЩИХ СИСТЕМ

Рассмотрены особенности построения информационно-измерительных и управляющих систем (ИИС) и целесообразность применения имитационного моделирования на начальных стадиях разработки ИИС. Представлена типовая структурная схема ИИС и приведены измеряемые физические параметры. Определен состав общего и специального программного обеспечения ИИС для автоматизированного рабочего места (АРМ) оператора, сервера и станции сбора данных. По результатам анализа типовой схемы и программного обеспечения предложен алгоритм построения и испытаний имитационной модели ИИС. Разработан алгоритм достижения цели операции от события к событию в заданный срок, в режиме реального времени. Получены адекватные оценки метрологических характеристик измерительных каналов, составных частей и ИИС в целом.

Основная тенденция развития измерений в автоматизированном производстве – это переход к машинному контролю по адаптивным моделям, к применению более сложных управляющих и информационно-измерительных систем (ИИС) [1].

Современный этап развития ИИС характеризуется переходом к созданию сложных иерархических систем, в которых осуществляется децентрализованная обработка информации, а отдельные части ИИС распределены в пространстве. Наблюдается тенденция к все большему использованию стандартных аппаратных и программных средств сопряжения на всех уровнях иерархии, что дает возможность создавать ИИС непосредственно у пользователя, исходя из его специфических требований.

ИИС можно определить как совокупность функционально связанных устройств, обеспечивающих измерение, сбор, вычислительную обработку и распределение измерительной информации в системе управления предприятием [2, 3].

Отличительной чертой ИИС является наличие в системе программируемой электронно-вычислительной машины (ЭВМ), которая используется не только для обработки результатов измерений, но и для управления процессом измерения, а также для управления воздействием (если это необходимо) на объект.

На рис. 1 представлена типовая структурная схема ИИС.



Рис. 1. Типовая структурная схема ИИС

Измерители первичные и нормализующие преобразователи должны осуществлять сбор и первичную обработки следующих физических параметров:

- давление (абсолютное, избыточное, динамическое);
- температуру (от минус 100 до 350 °С);
- силу (до 500 кН);
- перемещение (от 0,5 до 1250 мм);
- ускорение (от 0,5 до 10000 м/с^2, диапазон частот от 0,5 до 10000 Гц);
- расход (до 350 м^3/с);
- ток (до 200 А).

Станция сбора данных осуществляет сбор и хранение измерительной информации, поступающей от измерителей первичных и нормализующих преобразователей, и предоставляет данные на верхний уровень – АРМ оператора, сервер, обеспечивающий хранение данных и информации. С АРМ оператора информация поступает в локальную сеть предприятия [4].

Уровень развития ИИС определяется программным обеспечением (ПО) – совокупностью программ, используемых при подготовке и решении задач, управления и проверки работоспособности ИИС (рис. 2) [5, 6].

В соответствии с функциями, выполняемыми его компонентами, ПО ИИС состоит из двух видов – общего и специального ПО.

Общее ПО включает в себя ПО управления – совокупность программ, предназначенных для управления функционированием ИИС в процессе реализации прикладных измерительных программ и разработки новых программ.

Специальное ПО обычно реализуется в виде пакетов прикладных программ. Пакет прикладных программ – совокупность программ реализации стандартных методов (математической статистики, вычислительной математики, интеллектуального анализа данных) при решении различных прикладных задач.



Рис. 2. Состав программного обеспечения ИИС

В настоящее время отработка технических решений на начальных стадиях разработки ИИС на натурных макетах является длительным дорогостоящим процессом. В связи с развитием вычислительной техники все чаще используются методы имитационного моделирования, которое представляет собой процесс построения и испытания моделирующего алгоритма, имитирующего поведение и взаимодействие исследуемой ИИС с учетом случайных входных воздействий и внешней среды.

Имитационное моделирование целесообразно применять в тех случаях, когда:

- нет аналитических методов решения сформулированной задачи;
- аналитические методы есть, но не удовлетворяют метрологическим требованиям точности и достоверности;
- аналитические методы есть, но их вычислительные процедуры сложны даже для ЭВМ;
- реализация известных процедур сталкивается с недостаточной математической подготовкой исследователя;
- исследователю нужно знать не только оценки искомых характеристик, но и динамику всего процесса.

Одной из основных целей имитационного моделирования является определение показателей эффективности различных операций. Показатели эффективности могут выступать в виде оценок характеристик случайных величин, процессов или вероятностей исхода операций. В первом случае – это время, расход ресурсов, расстояния и т.п., во втором случае показатель эффективности выступает в качестве вероятности, например, достижения цели операции в заданный срок, исправного состояния техники и т.д. [7].

Исходными данными для определения метрологических характеристик ИИС служат метрологические характеристики компонентов: либо нормируемые в соответствии с [8], либо полученные экспериментально. Обычно известны (например, из паспорта на средство измерения) нормируемые метрологические характеристики, которые характеризуют воздействие факторов на результат измерительных процедур.

При учете воздействующих факторов приходится довольствоваться рядом упрощений и ограничений [9, 10]:

- ИИС рассматривается как совокупность независимых измерительных каналов;
- имитационными моделями отдельных составляющих погрешности являются либо детерминированные, либо случайные величины с равномерным или нормальным законами распределения, а также кусочно-линейные функции от них;
- функции распределения компонентов линейны;

• имитационными моделями измеряемых величин являются либо детерминированные величины, либо случайные процессы, удовлетворяющие условиям стационарных и динамических процессов;

• композиция составляющих погрешности имеет в общем случае аддитивно-мультипликативный характер;

• «алгоритмические» компоненты рассматриваются автономно, не в составе измерительных каналов;

• объектом рассмотрения (при определении метрологических характеристик средств измерений, но не погрешностей результата) служат лишь процедуры прямых измерений.

Эти обстоятельства обусловливают необходимость применения подходов, позволяющих снять подобные ограничения и получить более адекватные оценки метрологических характеристик [11].

В результате имитационного моделирования были решены следующие вопросы:

• разработана типовая структурная схема ИИС;

• проведен выбор общего и специального программного обеспечения;

• разработан алгоритм построения испытаний имитационной модели ИИС;

• разработан алгоритм достижения цели операции от события к событию в заданный срок в режиме реального времени;

• получены адекватные оценки метрологических характеристик составных частей и ИИС в целом.

Библиографический список

1. Парахуда, Р. Н. Информационно-измерительные системы : Письменные лекции / Р. Н. Парахуда, Б. Я. Литвинов. – СПб. : СЗТУ, 2002. – 74 с.

2. Древс, Ю. Г. Применение ПЭВМ в системах измерения и обработки данных. Ч. 2 Технические средства сбора и обработка информации в измерительно-вычислительных комплексах : учеб. пособие / Ю. Г. Древс, А. Л. Зорин. – М. : МИФИ, 2001.

3. Цапенко, М. П. Измерительные информационные системы: Структуры и алгоритмы, системотехническое проектирование : учеб. пособие для вузов / М. П. Цапенко. – 2-е изд., перераб. и доп. – М. : Энергоатомиздат, 1995. – 439 с.

4. Горшков, П. С. Функциональная модель процесса разработки автоматизированной системы контроля состояния авиационного оборудования / П. С. Горшков, А. В. Потемкин // Труды Международного симпозума. Надежность и качество. – 2016. – Т. 1. – С. 178–183.

5. Дмитриенко, А. Г. Распределенная интеллектуальная система мониторинга состояния объектов РКТ / А. Г. Дмитриенко, А. В. Блинов, А. В. Николаев // XXI век: итоги прошлого и проблемы настоящего. – 2011. – С. 15–21.

6. Козлов, В. В. Применение методов цифрового спектрального оценивания в задаче измерения параметров сигнала / В. В. Козлов, Б. В. Цыпин, М. Г. Мясникова, С. В. Ионов // Измерительная техника. – 2010. – № 10. – С. 16–20.

7. Дмитриенко, А. Г. Обобщенная процедура структурно-параметрического синтеза информационных моделей сложных систем / А. Г. Дмитриенко, М. Ю. Михеев, Т. В. Жашкова // XXI век: итоги прошлого и проблемы настоящего плюс. – 2012. – № 4. – С. 143–152.

8. ГОСТ 8.009–84. Нормирование и использование метрологических характеристик средств измерений.

9. МИ 222–80. Методика расчета метрологических характеристик измерительных каналов информационно-измерительных систем по метрологическим характеристикам компонентов.

10. РД 153-34.0-11.201–97. Методика определения обобщенных метрологических характеристик измерительных каналов ИИС и АСУ ТП по метрологическим характеристикам агрегатных средств измерений.

11. Климентьев, К. Е. Имитационное моделирование программно-управляемого процесса измерений / К. Е. Климентьев // Измерение. Мониторинг. Управление. Контроль. – 2012. – № 1. – С. 10–14.

А. В. Николаев, А. А. Папко, А. В. Поспелов

АО «Научно-исследовательский институт физических измерений», Пенза, Россия

МЕТОДЫ ПОВЫШЕНИЯ ВИБРАЦИОННОЙ ИНТЕЛЛЕКТУАЛЬНОЙ УСТОЙЧИВОСТИ

Представлены результаты разработки интеллектуальной системы устройств измерения углов тангажа и крена (УИТК), предназначенной для горизонтирования объектов, эксплуатируемых на подвижных шасси. Рассмотрены условия возникновения эффекта наложения спектров при эксплуатации УИТК и методы его исключения.

Горизонтирование при установке объектов различного назначения на подвижные шасси осуществляется с помощью систем с аналогичным названием.

Неотъемлемой частью систем горизонтирования являются инерциальные системы, предназначенные для измерения углов крена, тангажа и формирования управляющих воздействий для опорно-поворотных устройств установки объектов в заданное положение. При этом эффективность работы систем напрямую связана с точностью измерения углов тангажа и крена.

В качестве измерителей углов в наземных системах горизонтирования, как правило, используются акселерометры уравновешивающего преобразования [5]. В отличие от традиционных датчиков угла, использование акселерометров не требует механической связи между чувствительным элементом и подвижным основанием. Акселерометры измеряют углы наклона конструкций относительно вектора гравитационного ускорения g на основе энергетического взаимодействия с ним. При измерении углов тангажа θ и крена γ два акселерометра устанавливаются на платформу измерительными осями перпендикулярно. При этом они измеряют проекции вектора гравитационного ускорения, равные $g\theta = g \cdot \sin\theta$ и $g\gamma = g \cdot \sin\gamma$. Для каждого из них выходной сигнал без учета смещения нуля $U_{\text{вых}}(\theta)$ будет равен

$$U_{\text{вых}(\theta)} = \text{К}_{g\theta} \cdot \arcsin\frac{g_\theta}{g}\ , U_{\text{вых}(\gamma)} = \text{К}_{g\gamma} \cdot \arcsin\frac{g_\gamma}{g}\ ,$$

где К$g\theta$, К$g\gamma$ – коэффициенты преобразования акселерометров каналов θ и γ по гравитационному ускорению.

Особенностями создания акселерометров для систем горизонтирования является выполнение двух противоречивых требований:

– высокой точности, реализуемой за счет увеличения глубины уравновешивания и, как следствие, увеличения собственной частоты;

– устойчивости или сохранения метрологических характеристик при воздействии виброускорений в широком частотном диапазоне, возникающих при работе двигателя шасси, опорно-двигательных и технологических устройств объекта, например вентиляторов. При этом следует иметь в виду, что для систем горизонтирования с использованием акселерометров вибрационные помехи являются влияющей величиной, однородной с измеряемым параметром.

Выполнение указанных требований достигается путем формирования частотного диапазона измерений (ЧДИ), верхняя граница которого определяется контролируемой динамикой объекта и скоростью затухания амплитудно-частотной характеристики в первой октаве за верхней границей ЧДИ до 30 дБ/окт. При этом необходимость выполнения требований осложняется также тем, что параметры влияющих вибраций, как любой реальный сигнал, имеют неограниченный спектр, а требования к их частотному диапазону, как правило, задаются весьма приближенно.

В настоящее время для формирования ЧДИ акселерометров эффективно использование ФНЧ высокого порядка на переключаемых конденсаторах (на ПК-фильтрах). Для микросхем ПК-фильтров частота среза (верхняя граница ЧДИ) и частота генерации встроенного генератора, под которой следует понимать частоту дискретизации, связаны соотношением 50:1.

Структурная схема системы с учетом включения ПК-фильтров на входах АЦП микроконтроллера показана на рис. 1.

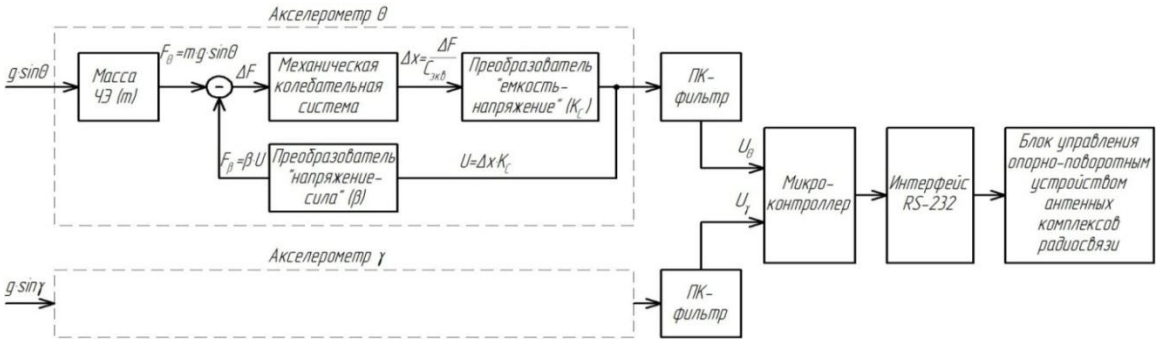


Рис. 1. Структурная схема системы горизонтирования

На рисунке обозначено:

g – модуль вектора гравитационного ускорения;

θ, γ – углы тангажа и крена соответственно;

m – масса инерционного элемента;

Δx – смещение инерционного элемента под действием измеряемой и уравновешивающей сил $F\theta$ ($F\gamma$), $F\beta$;

$C_{экв}$ – жесткость подвеса, формируемая электромеханической обратной связью.

Согласно структурной схеме проекции гравитационного ускорения на измерительные оси акселерометров каналов θ и γ преобразовываются в них в выходные напряжения, которые подаются на ПК-фильтры и входы АЦП микроконтроллера.

В настоящее время в АО «НИИФИ» разработана система горизонтирования типа УИТК. Ее отличительной особенностью является использование акселерометров собственной разработки с применением отечественной комплектации, включая ПК-фильтр, микроконтроллер и приемопередатчик, а также высокая стабильность метрологических характеристик, нормируемая в виде среднеквадратического значения абсолютной погрешности, которое в условиях эксплуатации не превышает ±20 угловых секунд (табл. 1). Для примера, эталонные средства горизонтирования типа поплавковых и электронных уровней имеют сопоставимую погрешность, но при неизменной температуре окружающей среды и отсутствии влияющих ускорений.

Таблица 1

Диапазоны измерений, град.	от ±3 до ±90
ЧДИ, Гц	0–4
Среднеквадратическое значение абсолютной погрешности в условиях эксплуатации, угл. с	±20
Время непрерывной работы, час, не более	15
Представление выходной информации	кодировка ASCII
Интерфейс	RS-232
Напряжение питания, В	23–34
Ток потребления, мА, не более	100
Температура окружающей среды, °С	–50 – +70
Виброускорение в направлениях измерительных осей, м/с^2/Гц	0,5/5–500
Виброускорение в поперечном направлении, м/с^2/Гц	5/5–500
Масса с корпусом из стали, кг	1,7
Габаритные размеры, мм	100×150×54

Несмотря на положительные результаты разработки и высокий уровень экспериментальной отработки УИТК, при ее пробной эксплуатации на подвижном шасси автомобиля «Тигр-М», системой записан спектр низкочастотного сигнала частотой до 2 Гц, который невозможно интерпретировать в качестве записей вибрации, воспроизведенной работой двигателя автомобиля. В режиме холостого хода автомобиля возможно наложение вибропроцессов от работы различных узлов двигателя, но их нижняя частотная граница, как правило, не бывает менее 300 Гц. Анализ вибрационных составляющих, возникающих при работе двигателя внутреннего сгорания [4, 6], показал, что причиной появления низкочастотной составляющей в спектре УИТК не может являться биение сигналов с незначительно отличающимися частотами, в связи с тем, что биение как линейный процесс не может быть источником новых частотных компонент.

Дальнейшие исследования показали, что появление в спектре выходного сигнала УИТК низкочастотной составляющей объясняется эффектом наложения спектров вибрационного сигнала и частоты дискретизации примененного ПК-фильтра. Указанный эффект характеризуется следующими признаками [1–3]:

– нарушением условия нахождения анализируемого сигнала (в рассматриваемом случае вибрационного) в зоне Найквиста или нарушением критерия дискретизации Котельникова, согласно которому частота дискретизации должна быть как минимум вдвое больше частоты исследуемого или влияющего сигнала;

– формированием в ПК-фильтре паразитного сигнала или низкочастотного образа реального сигнала, отображаемого в частотном диапазоне измерений системы и воспринимаемого АЦП и микроконтроллером в качестве измеряемого угла наклона (рис. 2).

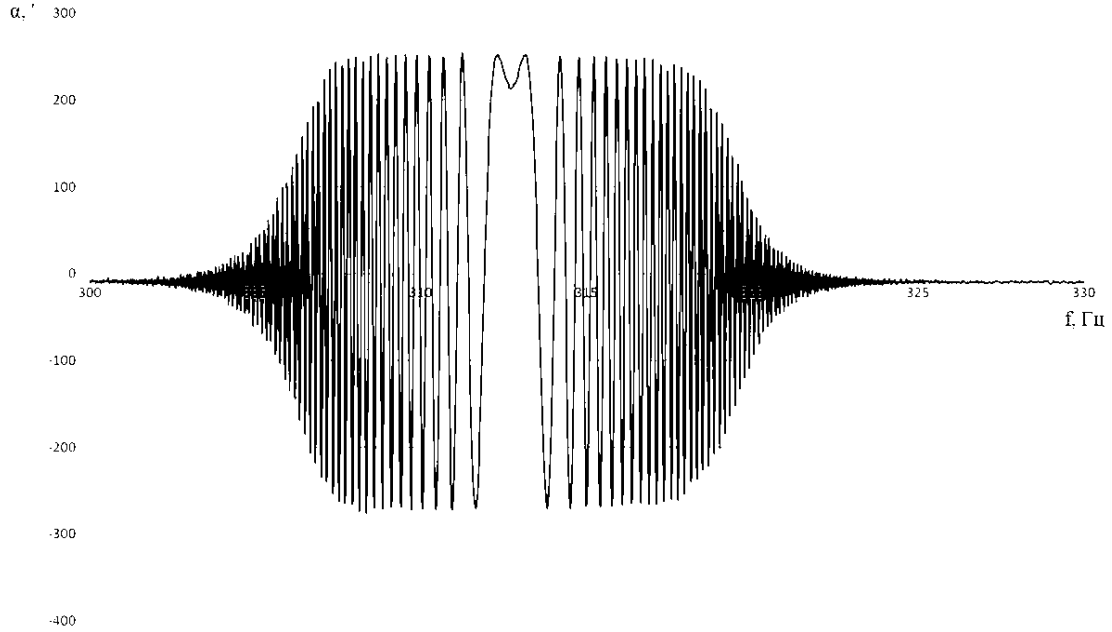


Рис. 2. Запись выходного сигнала УИТК при наложении спектров
с частотой воздействия вибрации, близкой к частоте дискретизации

В этой связи фактические измерения низкочастотной составляющей при установке УИТК на автомобиль «Тигр-М» объясняются отображением частот выше частоты дискретизации ПК-фильтра в полосу частот полезного сигнала, находящуюся в диапазоне частот от 0 до верхней границы ЧДИ.

Из анализа признаков наложения спектров, а также учитывая, что эффект наложения в цифровом сигнале полностью производится на первом этапе цифро-аналогового преобразования, а в данном случае – в ПК-фильтре, и то, что для любого реального сигнала спектр является бесконечным, следует, что для его исключения при работе УИТК необходимо принятие специальных мер.

К их числу относятся:

– увеличение частоты дискретизации ПК-фильтра, что связано с нежелательным расширением верхней границы ЧДИ;

– использование более предпочтительного решения в виде ограничения спектра измеряемого сигнала путем дополнительной аналоговой фильтрации выходного сигнала акселерометров.

Реализация этого решения и его экспериментальное подтверждение позволили решить достаточно сложную проблему наложения спектра включением пассивного интегрирующего звена первого порядка на вход ПК-фильтра с частотой среза, превышающей верхнюю границу ЧДИ в (5−10) раз.

Таким образом, для правильной интерпретации результатов измерений систем горизонтирования на основе акселерометров в процессе эксплуатации необходимо не только фильтровать анализируемый сигнал до начала дискретизации, но и обращать особое внимание на достоверность представления частотного диапазона вибрационных помех, возникающих при работе шасси, опорно-поворотных и технологических устройств объектов горизонтирования.

Кроме этого представляется целесообразным нормирование частоты наложения спектров в КД системы на основе экспериментального исследования при воздействии влияющих вибраций в интервале частот, верхняя граница которого превышает на порядок максимальную частоту вибраций, установленную в техническом задании.

Библиографический список

1. Автомобили семейства «Тигр». Виброакустический метод диагностики подшипников коленвала двигателей внутреннего сгорания. Вибрация двигателя на холостых // Техническая документация.

2. ГОСТ 32108−2013 Вибрация. Измерения вибрации, передаваемой машиной через упругие изоля-торы. Двигатели внутреннего сгорания поршневые высокоскоростные и среднескоростные. – URL: docs.cntd.ru

3. ГОСТ Р ИСО 13373-2-209 Контроль состояния и диагностика машин. Вибрационный контроль состояния машин. Ч. 2. Обработка, анализ и представление результатов измерений вибрации. – URL: docs.cntd.ru

4. Куприянов, М. С. Цифровая обработка сигналов / М. С. Куприянов, Б. Д. Матюшкин. – 2-е изд. – СПб. : Политехника, 2000. – 592 с.

5. Мокров, Е. А. Статико-динамические акселерометры для ракетно-космической техники / Е. А. Мокров, А. А. Папко. – Пенза : ПАИИ, 2004. – 164 с.

6. Лайонс, Р. Цифровая обработка сигналов / Ричард Лайонс. – 2-е изд. – М. : Бином-Пресс, 2006.

В. Н. Тарасов, Н. Ф. Бахарева, Л. В. Липилина

Поволжский государственный университет телекоммуникаций
и информатики, Самара, Россия

МОДЕЛИРОВАНИЕ ТЕЛЕТРАФИКА В СЛУЧАЕ ШИРОКОГО ДИАПАЗОНА ИЗМЕНЕНИЯ ЕГО ПАРАМЕТРОВ НА ОСНОВЕ ТЕОРИИ МАССОВОГО ОБСЛУЖИВАНИЯ

Представлено исследование систем массового обслуживания (СМО) G/G/1 с произвольными входными распределениями с коэффициентами вариаций интервала между поступлениями требований и времени обслуживания меньшими и большими 1, т.е. перекрывающими весь интервал (0, ∞). Для этого рассмотрены системы M/M/1 с запаздыванием во времени и $H_2/H_2/1$ с гиперэкспоненциальными входными распределениями.

Введение. Статья посвящена исследованию систем массового обслуживания (СМО) G/G/1 с произвольными входными распределениями с коэффициентами вариаций интервала между поступлениями требований и времени обслуживания меньшими и большими 1, т.е. перекрывающими весь интервал (0, ∞). Для этого рассмотрены системы M/M/1 с запаздыванием во времени [1] и $H_2/H_2/1$ с гиперэкспоненциальными входными распределениями [2–4] с целью получения решения для среднего времени ожидания требований в очереди. Для этого использован метод спектрального разложения решения интегрального уравнения Линдли. Для практического применения полученных результатов использован метод моментов. В работе [1] показано, что коэффициент вариации случайной величины, распределенной по экспоненциальному закону с запаздыванием во времени, меньше 1.

Известно, что распределенная по гиперэкспоненциальному закону H_2 случайная величина имеет коэффициент вариации *c* больше 1. Учитывая тот факт, что распределение H_2 является трехпараметрическим, в статье приведен механизм аппроксимации произвольных законов распределений гиперэкспоненциальным как на уровне двух первых моментов, так и на уровне трех первых моментов.

Решение для системы M/M/1 с запаздыванием (случай $c_\lambda < 1, c_\mu < 1$)

Рассмотрим систему массового обслуживания (СМО), на вход которой поступают требования, случайные интервалы между которыми распределены по закону [1]:

$$a(t) = \begin{cases} \lambda e^{-\lambda(t-t_0)}, & t \ge t_0 \\ 0, & 0 \le t < t_0 \end{cases}. \qquad (1)$$

Аналогично распределено и время обслуживания:

$$b(t) = \begin{cases} \mu e^{-\mu(t-t_0)}, & t \ge t_0 \\ 0, & 0 \le t < t_0 \end{cases}. \qquad (2)$$

Функции плотности (1) и (2) являются сдвинутыми вправо от нулевой точки на величину t_0 экспоненциальными распределениями с двумя параметрами (λ, t_0) и (μ, t_0), причем $\lambda < \mu$. Таким образом, имеем СМО с запаздыванием во времени на величину [1].

В работе [1] приведены преобразование Лапласа функции плотности времени ожидания, полученное методом спектрального разложения решения интегрального уравнения Линдли: $W^*(s) = s\Phi_+(s) = \dfrac{(1-\lambda/\mu)(\mu+s)}{(s+\mu-\lambda)}$, и полученное через него среднее время ожидания

$$\bar{W} = \frac{\lambda/\mu}{\mu-\lambda}. \tag{3}$$

Для определения неизвестных параметров (3) получена система уравнений:

$$\begin{cases} \lambda^{-1} + t_0 = \bar{\tau}_\lambda \\ (1+\lambda t_0)^{-1} = c_\lambda \\ \mu^{-1} + t_0 = \bar{\tau}_\mu \\ (1+\mu t_0)^{-1} = c_\mu \end{cases}, \tag{4}$$

где числовые характеристики в правых частях системы: средние значения интервалов $\bar{\tau}_\lambda$ и $\bar{\tau}_\mu$, коэффициенты вариации c_λ и c_μ меньшие единицы при $t_0, \lambda, \mu > 0$, будут входными параметрами для определения неизвестных параметров. В связи с этим мы имеем немарковскую модель массового обслуживания. Среднее время ожидания требования в очереди в такой системе будет меньше, чем в системе M/M/1 при одинаковом коэффициенте загрузки. Кроме того, использование функций плотности (1) и (2) позволяет аппроксимировать исходные входные распределения на уровне двух первых моментов, в отличие от классической системы M/M/1.

В качестве примера рассмотрим случай высокой нагрузки при параметрах: $\bar{\tau}_\mu = 1$, $\bar{\tau}_\lambda = 10/9$ и $c_\mu = 0.5$. В этом случае получим загрузку $\rho = 0{,}9$, значение $c_\lambda = 0{,}55$, а из (4) значения $\lambda = 18/11$ и $\mu = 2$.Тогда среднее время ожидания по формуле (3) $\bar{W} = \dfrac{18/11/2}{2-18/11} = \dfrac{18\cdot 11}{4\cdot 22} = \dfrac{9}{4}$.

Система M/M/1 при той же нагрузке дает $\bar{W} = \dfrac{0.9/1}{1-0.9} = 9$, т.е. в 4 раза большую задержку. Таким образом, рассмотренная система массового обслуживания с запаздыванием позволяет рассчитать ее характеристики при коэффициентах вариации интервалов между поступлениями требований c_λ и времени обслуживания c_μ, меньших 1 при некоторых ограничениях на входные параметры системы.

Решение для системы $H_2/H_2/1$ (случай $c_\lambda \ge 1, c_\mu \ge 1$).

В работах [2, 4] авторами найдено преобразование Лапласа-Стилтьеса для функции плотности времени ожидания для системы $H_2/H_2/1$:

$$W^*(s) = \frac{s_1 s_2 (s+\mu_1)(s+\mu_2)}{\mu_1\mu_2 (s+s_1)(s+s_2)}, \tag{5}$$

где $-s_1$ и $-s_2$ отрицательные вещественные части корней кубического уравнения

$$s^3 - c_2 s^2 - c_1 s - c_0 = 0. \tag{6}$$

Коэффициенты этого уравнения: $c_0 = a_0 b_1 - a_1 b_0 - a_0(\mu_1+\mu_2) + b_0(\lambda_1+\lambda_2)$, $c_1 = -a_1 b_1 - a_0 - b_0 + (\lambda_1+\lambda_2)(\mu_1+\mu_2)$, $c_2 = \lambda_1+\lambda_2-\mu_1-\mu_2$, а параметры: $a_0 = \lambda_1\lambda_2$, $a_1 = p\lambda_1 + (1-p)\lambda_2$, $b_0 = \mu_1\mu_2$, $b_1 = q\mu_1 + (1-q)\mu_2$. Величины $p, \lambda_1, \lambda_2, q, \mu_1, \mu_2$ являются параметрами гиперэкспоненциальных распределений с функциями плотностей

$$a(t) = p\lambda_1 e^{-\lambda_1 t} + (1-p)\lambda_2 e^{-\lambda_2 t}, \quad (7)$$

$$b(t) = q\mu_1 e^{-\mu_1 t} + (1-q)\mu_2 e^{-\mu_2 t}. \quad (8)$$

соответственно для системы $H_2/H_2/1$.

Среднее время ожидания в очереди равно значению производной от функции преобразования Лапласа (5) со знаком минус в точке s=0. Окончательно, среднее время ожидания в очереди для СМО $H_2/H_2/1$:

$$\overline{W} = \frac{1}{s_1} + \frac{1}{s_2} - \frac{1}{\mu_1} - \frac{1}{\mu_2}. \quad (9)$$

Выражение (5) на основе свойств преобразования Лапласа, позволяет также определить моменты высших порядков для времени ожидания. Например, начальный момент 2-го порядка времени ожидания равен значению второй производной от преобразования (5) в точке $s = 0$ и он будет иметь вид:

$$\overline{W^2} = \frac{d^2 W^*(s)}{ds^2} = \frac{2}{\mu_1\mu_2} - \frac{2}{s_1 s_2} - \frac{2(s_1+s_2)}{\mu_1\mu_2 s_1^2 s_2^2} \times [s_1 s_2(\mu_1+\mu_2) - \mu_1\mu_2(s_1+s_2)]. \quad (10)$$

В свою очередь, начальный момент 2-го порядка позволяет определить дисперсию времени ожидания. Учитывая определение джиттера в телекоммуникациях как разброс времени ожидания от его среднего значения, тем самым получим возможность определения джиттера через дисперсию. Это является важным результатом для анализа трафика, чувствительного к задержкам.

Для практического применения результатов (9) и (10) необходимо определить входящие в них параметры. Определение этих неизвестных параметров рассматривается ниже.

Аппроксимация законов распределений на уровне двух первых моментов

Воспользуемся свойством преобразования Лапласа воспроизведения моментов и запишем начальные моменты до второго порядка для распределения (7):

$$\overline{\tau}_\lambda = \frac{p}{\lambda_1} + \frac{(1-p)}{\lambda_2}, \quad (11)$$

$$\overline{\tau_\lambda^2} = \frac{2p}{\lambda_1^2} + \frac{2(1-p)}{\lambda_2^2}. \quad (12)$$

Аналогично, выписывая начальные моменты до 3-го порядка включительно, можно выполнить аппроксимацию гиперэкспоненциального закона распределения на уровне трех моментов, что очевидно, является более сложной задачей. Рассматривая равенства (11) и (12) как уравнения метода моментов, найдем неизвестные параметры распределения (7) λ_1, λ_2, p. Для этого запишем связующее условие в виде выражения для квадрата коэффициента вариации

$$c^2 = \frac{\overline{\tau_\lambda^2} - (\overline{\tau}_\lambda)^2}{(\overline{\tau}_\lambda)^2}. \quad (13)$$

Заметим, что система уравнений (11) и (12) с тремя неизвестными является недоопределенной. Исходя из вида уравнения (11) положим

$$\lambda_1 = 2p/\overline{\tau}_\lambda, \quad \lambda_2 = 2(1-p)/\overline{\tau}_\lambda \quad (14)$$

и потребуем выполнения условия (13). Подставив выражения (11), (12) и (14) в (13) и решив квадратное уравнение относительно параметра p, получим для него два значения:

$p = \frac{1}{2}(1 \pm \sqrt{\frac{c_\lambda^2 - 1}{c_\lambda^2 + 1}})$. При этом можно воспользоваться любым из них. Аналогично поступим с распределением (8). Таким образом, гиперэкспоненциальный закон расп-ределения может определяться полностью двумя первыми моментами и перекрывать весь диапазон изменения коэффициента вариации от 1 до ∞.

Рассмотрим пример. Пусть коэффициент загрузки СМО $\rho = \bar{\tau}_\mu / \bar{\tau}_\lambda = 0{,}9$, где $\bar{\tau}_\lambda$ и $\bar{\tau}_\mu$ средние значения интервалов между поступлениями и времени обслуживания. Рассмотрим случай нормированного обслуживания $\bar{\tau}_\mu = \mu^{-1} = 1$. Тогда средний интервал между поступлениями $\bar{\tau}_\lambda = 10/9.$ Пусть коэффициенты вариаций случайных величин – интервалов между поступлениями и времени обслуживания $c_\lambda = c_\mu = 2$. Аппроксимация на уровне двух первых моментов дает: $p \approx 0{,}887$, $\lambda_1 \approx 1{,}597$, $\lambda_2 \approx 0{,}203$, $q \approx 0{,}887$, $\mu_1 \approx 1{,}775$, $\mu_2 \approx 0{,}225$. Таким образом, неизвестные параметры распределений (7) и (8) однозначно определены. Теперь воспользуемся результатом для системы $H_2/H_2/1$ (9), приведенным выше. Коэффициенты кубического уравнения (6) в этом примере равны: $c_0 \approx 0{,}014$; $c_1 \approx 0{,}572$; $c_2 = -0{,}20$. Найдем корни кубического уравнения (8) с помощью пакета Mathcad: $-s_1 \approx -0{,}852$, $-s_2 \approx -0{,}025$, $s_3 \approx 0{,}677$. Тогда среднее время ожидания равно $\overline{W} \approx 36{,}20$.

Заключение

Полученные результаты приводят к следующим выводам.

1. В том случае, когда коэффициенты вариации интервалов поступления и времени обслуживания меньше 1, можно использовать систему M/M/1 с запаздыванием.

2. В случае, если коэффициенты вариации интервалов поступления и времени обслуживания больше 1, можно использовать систему $H_2/H_2/1$.

3. Рабочие формулы для расчета задержки в обоих случаях, приведены выше.

Библиографический список

1. Тарасов, В. Н. Анализ и расчет системы массового обслуживания с запаздыванием / В. Н. Тарасов, Н. Ф. Бахарева, И. А. Блатов // Автоматика и телемеханика. – 2015. – № 11. – С. 51–59.

2. Тарасов, В. Н. Исследование систем массового обслуживания с гиперэкспоненциальными входными распределениями / В. Н. Тарасов // Проблемы передачи информации. – 2016. – № 1. – С. 16–26.

3. Тарасов, В. Н. Математическая модель трафика с тяжелохвостным распределением на основе системы массового обслуживания H_2/M/1 / В. Н. Тарасов, Н. Ф. Бахарева, Г. А. Горелов // Инфокоммуникационные технологии. – 2014. – № 3. – С. 36–41.

4. Тарасов, В. Н. Определение среднего времени ожидания требований в управляемой системе массового обслуживания $H_2/H_2/1$ / В. Н. Тарасов, И. В. Карташевский // Системы управления и информационные технологии. – 2014. – Т. 57, № 3. – С. 92–96.

5. Тарасов, В. Н. Анализ входящего трафика на уровне трех моментов распределений временных интервалов / В. Н. Тарасов, Н. Ф. Бахарева, Г. А. Горелов, С. В. Малахов // Информационные технологии. – 2014. – № 9. – С. 54–59.

Mustafa Sadeq Jaafar, Sergey A. Zinkin, Dmitry V. Pashchenko

Penza State University, Penza, Russia

RECONFIGURABLE NETWORK MODELS FOR DISTRIBUTED COMPUTING SYSTEMS

The article is devoted to the development of customizable software for distributed computing systems. In order to develop such intermediate-level (middleware) software, it is suggested to use modifiable, or changeable, models based on Petri nets. The basis for constructing reconfigurable Petri nets (RPN – reconfigurable, or changeable, Petri nets) is the ordinary, lively and safe (1-bounded) Petri nets, which have no configuration restrictions and the multiplicity of arcs in which does not exceed one. The proposed approach is illustrated by an example of a simple protocol for peer-to-peer computing systems, where the roles of clients and servers can change during the process of the distributed system work.

1. Introduction

The size and complexity of the hardware and software of computing systems are noticeably increasing. The design of computer networks, distributed computing systems, including cloud and grid systems, wireless mobile networks and other such objects, forces developers to look for new approaches to the design of continuously modifiable hardware and software.

The newest technologies for the design of distributed systems are based, essentially, on the concept of "network is a computer", promoted by Sun Microsystems and later used by Oracle Corporation. This concept was further developed in the form of a paradigm for cloud applications that have the properties of “multilease” and “live” database migration. A particular interest in this concern is the development of a middleware for large reconfigurable clustered server systems as part of supporting the “big data” concept. Therefore, there was considerable interest in designing a reconfigurable and parametrically tuned functional architecture of distributed computing systems. In some cases, the implementation of this approach can increase the efficiency and reduce the cost of large software and hardware systems.

2. Reconfigurable and binary controlled Petri nets and some of their interpretations

The basis for constructing reconfigurable Petri nets (RPN – reconfigurable, or changeable, Petri nets) is the ordinary, lively and safe (1-bounded) Petri nets, which have no configuration restrictions and the multiplicity of arcs in which does not exceed one.

The basic information about liveliness, safety and good formation of the Petri nets are consistent with those adopted in the basic research papers [1, 2]. The interpretation of the RPN in the future will be based for the production and logical-algebraic representations.

In [2] it was stated that Petri nets are a promising tool for describing and studying information processing systems that are characterized as being concurrent, asynchronous, distributed, parallel, nondeterministic, and/or stochastic. The fundamental properties of Petri nets were also considered in paper [3]. Problems and exercises related to reachability graph, boundedness, coverability graph, liveness, reversibility, persistence and fairness, conservativeness, P-invariants and T-invariants, concurrency and conflict, analysis of Petri net properties, structural properties of Petri nets, are given in article [4].

Article [5] deals with the supervisory control of concurrent systems based on a Petri net structural approach. This article addresses the formal development of tools that design supervisors automatically, given a Petri net model and a specification. The article also addresses the problems of deadlock prevention and liveness properties of Petri nets.

In paper [6], the authors review the concepts of modification in Petri nets with respect to their capabilities of modeling self-organizing, mobile communication networks and further they stated that only a few approaches, however, reflect the ability to adopt their structure to their current dynamics; this property is becoming more and more important regarding modern communication networks. Self-modifying nets (SMnets) in [7] basically are place-transition nets (PTNs) extended by marking-dependent arc-cardinality. Colored Petri nets with changeable structure were considered in [8]. The authors of this work considered Temporized Colored Petri nets with changeable structure (TCPN-CS). A TCPNCS consists of a sequence of TCPNs along with a mechanism to alter structure according to this sequence.

Controlled Petri nets (CtlPNs) considered in [9] are a class of Petri nets with external enabling conditions called control places which allow an external controller to influence the progression of tokens in the net. CtlPNs were first introduced by Krogh [10] and Ichikawa and Hiraishi [11].

The authors of article [12] proposed the similar concept of the modified binary Petri net. The idea is to follow the traditional Petri net firing mechanisms, but with a restriction that no places can be marked with more than one token. An algorithm was designed to capture transition firings based on the state equation and the priority assignment. But in the considered networks, information and inhibitory arcs are not used.

Some varieties of binary Petri nets are also considered in [13]. Article [14] confirms that modeling the adaption of a system to a changing environment gets more and more important. Application areas cover e.g. computer supported cooperative work, multi agent systems, dynamic process mining or mobile networks. The main idea of Petri net transformations in [14] is to extend the classical theory of Petri nets by a rule-based technique that allows to model the changes of the Petri net structure.

As an illustrative example, a simple Petri net (figures 1, 2), which describes well-known interactions of the nodes in the computer network at the applied and transport levels, is next chosen. Such an illustrative example was considered earlier in a number of researches, for example, in [2, 14, 15, 16].

However, in this paper, this example was used to construct a special type of Petri net – RPN, whose graphical representation contains duplicate positions. The work of such a Petri net is typical for peer-to-peer systems.

3. Reconfigurable network models for distributed computing systems

The construction of the proposed kind of Petri nets is also based on a structural and reconfigurable approach, differing, however, taking into account the structural links and supervisory control. Here figure 1 and figure 2 show the same RPN network that simulates client-server interactions in the computer network. Especially note that both figures represent the same network. In each part of this network, the same positions can be located. A feature of this RPN network is that it allows one of two implementations (defined and used in the present work, the RPN networks in principle admit a set of implementations that can be established dynamically, during the operation of the network, for example, after the operation of certain transitions).

In the implementation of the model shown in figure 1, the RPN is configured so that the initiator of message exchanges is a client application described by the positions a_1, a_2, a_3, a_4, a_5 and transitions t_1, t_2, t_3, t_4, t_5. The server application is represented in the model by the positions b_1, b_2, b_3, b_4, b_5 and the transitions u_1, u_2, u_3, u_4, u_5.

Intermediate positions represent message-sending and message-receiving buffers mes_1 (from client to server) and mes_2 (from server to client). The remaining intermediate positions act as buffers for ack_1 acknowledgment messages (from the server to the client) and ack_2 (from the client to the server).

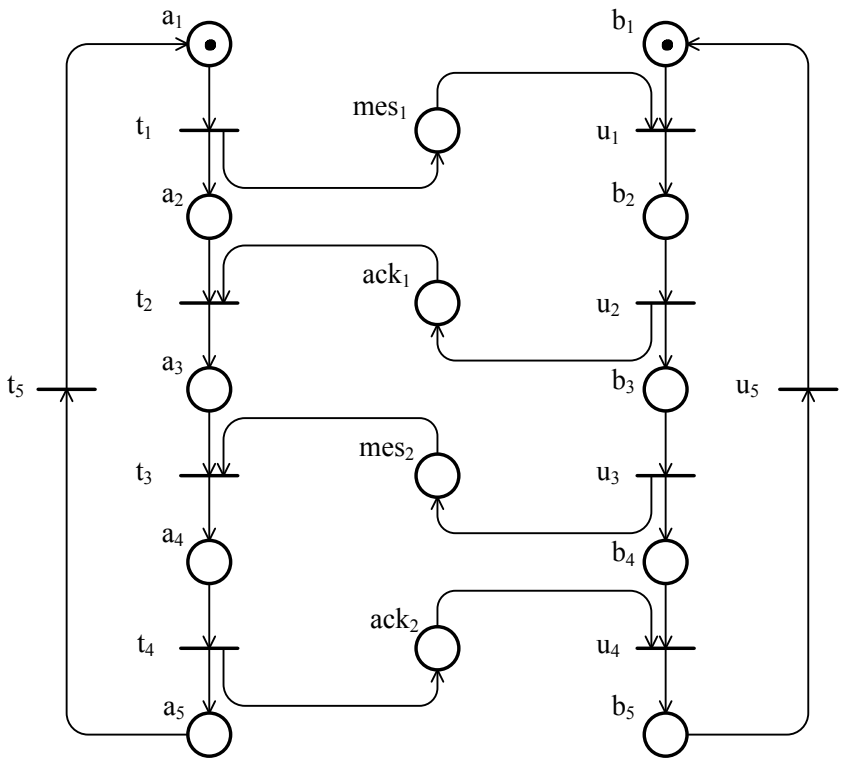


Fig. 1. Reconfigurable Petri net (RPN: part 1): implementation for the case where the initiator of the dialog is the client application

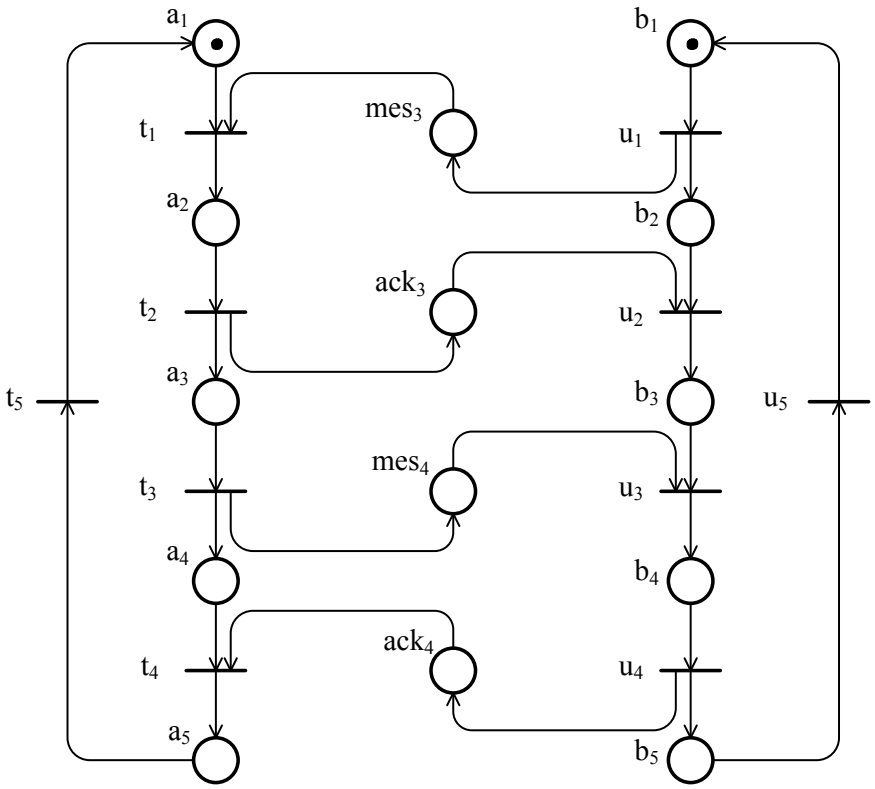


Fig. 2. Reconfigurable Petri net (RPN: part 2): implementation for the case where the initiator of the dialog is the application server

Suppose that after changing the operation mode of the client-server system, the initiator of the messaging should be the server. Such changes in the operation modes of clients and servers are typical, for example, for peer-to-peer subsystems in computer networks. The work of the system in this case is determined by the implementation of the RPN model shown in figure 2. This work assumes that the structure of both the living and safe RPN under consideration without multiple arcs is determined by two predicates $F(p, t)$ and $H(t, p)$ known in Petri net theory:

$$F: P \times T \to \{\textbf{true}, \textbf{false}\};$$
$$H: T \times P \to \{\textbf{true}, \textbf{false}\}.$$

For example, the truth of the statement $F(p_i, t_i)$ means that there is an arc that joins the position p_i with the transition t_i , and the truth of the statement $H(t_m, p_n)$ means that there is an arc connecting the transition t_m to the position p_n.

Depending on the given predicates, the RPN can be configured to one of two implementations, corresponding to the selected messaging mode in the client-server system.

The liveliness and safety of RPN can be established by known methods using some tools for analyzing Petri nets.

Now the task of designing a parametrically customizable application in a client-server architecture, in which the client and the server can change roles. One approach to solve this problem in which two pairs of independent client-server applications are created. One pair of applications "client" and "server" can correspond to the Petri net in figure 1, and the second, independent from the first one, can correspond to the Petri net in figure 2.

In this paper, a different approach based on the reconfiguration of one Petri net will be used. Noticing that in figure 1 and figure 2 which are showing two implementations of the RPN network, in which only the positions mes_1, ack_1, mes_2, ack_2, or positions mes_3, ack_3, mes_4, ack_4 are used. This means that the client and the server can change roles in the computer network. This approach is beneficial in the case when the majority of the particular Petri net during the reconfiguration remains unchanged.

4. Deadlock avoidance for Petri net models

Matlab based analysis of Petri nets designed in Pmediteur in [16] is done by the procedure of deadlock avoidance for a Petri Net model like in figures 1, 2 of two communicating computers in deadlock situation. Mathematical methods for calculating invariants in Petri nets and the graph of reachable markings are used for analyzing the properties of bounded net. Functions of Matlab Toolbox 1.0 can be downloaded from web site http://dce.felk.cvut.cz/cak/research/PN/Matlab_Toolbox.

Figures 3, 4 are showing Petri Net models (similar to the pictures in [16]) of two communicating computers in a deadlock situation; the problem is that both computers are ready to receive but none is ready to send. For the Petri net in figure. 3 the deadlock situation occurs when the statements $P(a_2)$ and $P(b_1)$ are true. Similarly, for the Petri net in figure. 3 the deadlock situation occurs when the statements $P(a_3)$ and $P(b_4)$ are true. Here P: $Pos \to \{\textbf{true}, \textbf{false}\}$ is an unary marking function (unary predicate) defined for a 1-bounded fragment of the Petri net, Pos is a finite set of places.

Through the growing popularity of using formal tools based on Petri nets similar to [16], when creating prototypes of distributed computing systems on the hardware and software platforms of both the local and global networks, the implementation of such tools has found wide distribution, as was indicated by an overview of known tools from the site http://www.informatik.uni-hamburg.de/TGI/PetriNets/tools/quick [17].

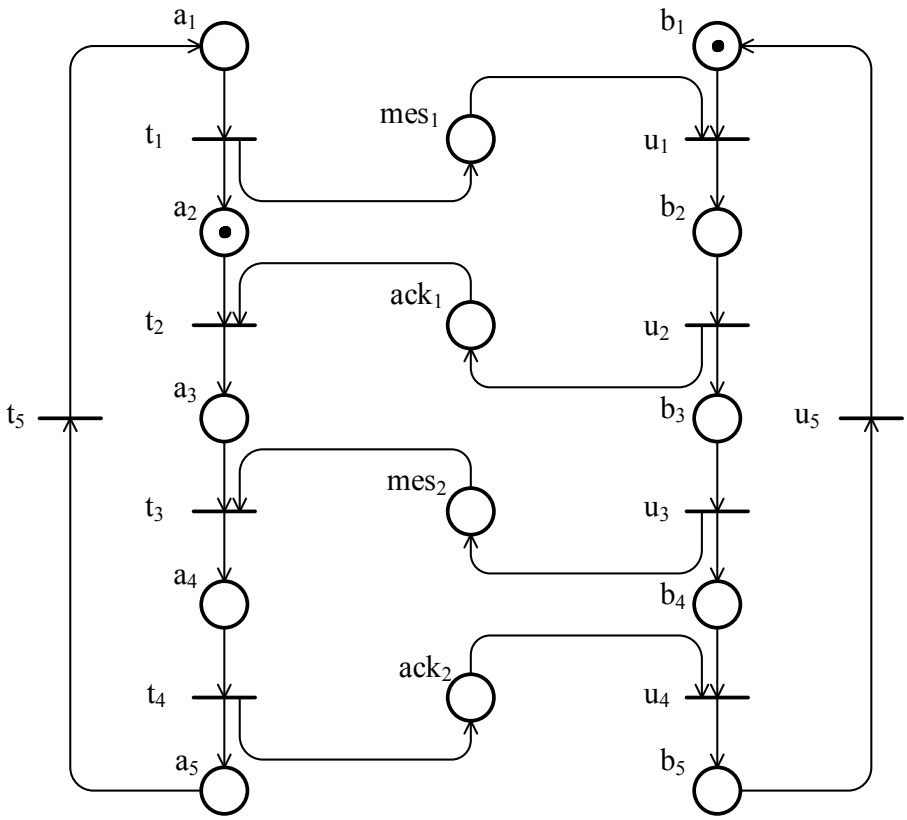


Fig. 3. Deadlock situation for two communicating computers (case 1)

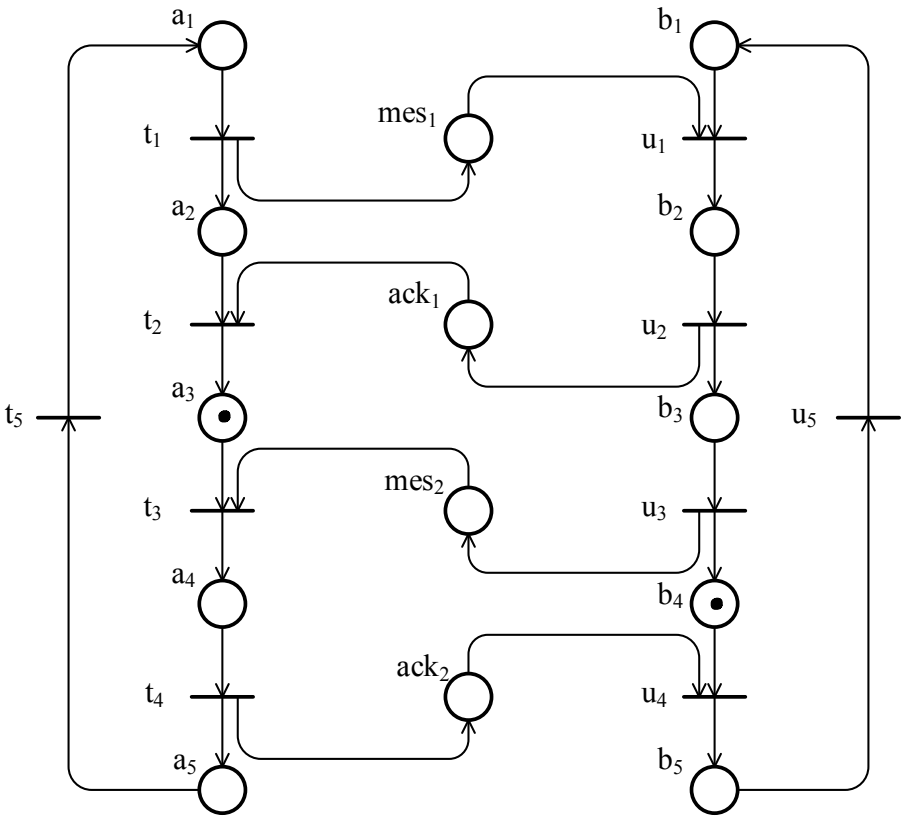


Fig. 4. Deadlock situation for two communicating computers (case 2)

5. Parametric adjustment of the production rules

Synthesis of distributed applications is proposed to be carried out on the basis of formalized specifications, which are used as logical production rules for firing transitions. In contrast to the known product rules, in new rules the statements of the form $F(p_i, t_i)$ and $H(t_m, p_n)$ reflecting the structure of the Petri net will be used. It becomes possible, when changing the truth of such statements, to translate the Petri net from one mode of functioning to another. At the same time, considerable flexibility of the transformations is provided, they do not affect the production rules, and the choice of a rule is determined by the changeable structural links of the transitions and positions in the particular Petri net. For the first mode of operation (it corresponds to the implementation of the RPN network in figure 1), the "parametrically tunable" R_1 transition rules, or the rules of logical inference, are presented in the following form:

System of parametrically tunable product rules R_1:

t_1: $mt_1 = F(a_1, t_1)\&H(t_1, a_2)\&H(t_1, mes_1)\&P(a_1) \to \neg P(a_1)\&P(a_2)\&P(mes_1)$;

u_1: $mu_1 = F(b_1, u_1)\&F(mes_1, u_1)\&H(u_1, b_2)\&P(mes_1)\&P(b_1) \to \neg P(mes_1)\& \neg P(b_1)\&P(b_2)$;

t_2: $mt_2 = F(a_2, t_2)\&F(ack_1, t_2)\&H(t_2, a_3)\&P(a_2)\&P(ack_1) \to \neg P(a_2)\&\neg P(ack_1)\&P(a_3)$;

u_2: $mu_2 = F(b_2, u_2)\&H(u_2, ack_1)\&H(u_2, b_3)\&P(b_2) \to \neg P(b_2)\&P(ack_1)\&P(b_3)$;

t_3: $mt_3 = F(a_3, t_3)\&F(mes_2, t_3)\&H(t_3, a_4)\&P(a_3)\&P(mes_2) \to \neg P(a_3)\&\neg P(mes_2)\&P(a_4)$;

u_3: $mu_3 = F(b_3, u_3)\&H(u_3, mes_2)\&H(u_3, b_4)\&P(b_3) \to \neg P(b_3)\&P(mes_2)\&P(b_4)$;

t_4: $mt_4 = F(a_4, t_4)\&H(t_4, a_5)\&H(t_4, ack_2)\&P(a_4) \to \neg P(a_4)\&P(a_5)\&P(ack_2)$;

u_4: $mu_4 = F(ack_2, u_4)\&F(b_4, u_4)\&H(u_4, b_5)\&P(b_4)\&P(ack_2) \to \neg P(b_4)\&\neg P(ack_2)\&P(b_5)$;

t_5: $mt_5 = F(a_5, t_5)\&H(t_5, a_1)\&P(a_5) \to \neg P(a_5)\&P(a_1)$;

u_5: $mu_5 = F(b_5, u_5)\&H(u_5, b_1)\&P(b_5) \to \neg P(b_5)\&P(b_1)$.

For example, a meaningful description of the production rule mt_1 for the transition t_1 would be like this: "if the position a_1 is the input for the transition t_1, the positions a_2 and mes_1 are the outputs for the same transition, and at position a_1 there is a marker, then from the position a_1, this marker is deleted, and in each of the positions a_2 and mes_1 appears one marker".

Speaking in a formal language, if before the firing of the transition t_1, a compound statement below was true:

$$F(a_1, t_1)\&H\,(t_1, a_2)\&H(t_1, mes_1)\&P(a_1),$$

Then after firing this transition, the following compound statement will be true:

$$\neg P\,(a_1)\&P(a_2)\&P(mes_1).$$

In a real client-server system, firing of t_1 corresponds to the transmission of message mes_1 from the client application to the server application through the corresponding logical and physical I/O ports. The production rules for firing transitions are interpreted, and the corresponding software modules are then executed in a networked environment.

For the second mode of operation (it corresponds to the implementation of the RPN network in figure 2), the "parametrically tunable" transition rules R_2, or the rules of logical inference, are presented below. The asterisks here indicate alternative rules for the firing of RPN network transitions:

The system of parametrically tunable product rules R_2:

t_1: $mt_1^* = F(a_1, t_1)\&F(mes_3, t_1)\&H(t_1, a_2)\&P(a_1)\&P(mes_3) \to \neg P\,(a_1)\&\neg P\,(mes_3)\&P(a_2)$;

u_1: $mu_1^* = F(b_1, u_1)\&H(u_1, mes_3)\&H(u_1, b_2)\&P(b_1) \to \neg P(b_1)\&P(mes_3)\&P(b_2)$;

t_2: $mt_2^* = F(a_2, t_2)\&H(t_2, a_3)\&H(t_3, ack_3)\&P(a_2) \to \neg P\,(a_2)\&P(a_3)\&P(ack_3)$;

u_2: $mu_2^* = F(b_2, u_2)\&F(ack_3, u_2)\&H(u_2, b_3)\&P(ack_3)\&P(b_2) \to \neg P(ack_3)\&\neg P(b_2)\&P(b_3)$;

t_3: $mt_3^* = F(a_3, t_3)\&H(t_3, a_4)\&H(t_3, mes_4)\&P(a_3) \to \neg P(a_3)\&P(a_4)\&P(mes_4)$;

u_3: $mu_3^* = F(b_3, u_3)\&F(mes_4, u_3)\&H(u_3, b_4)\&P(mes_4)\&P(b_3) \to \neg P(mes_4)\&\neg P(b_3)\&P(b_4)$;

t_4: $mt_4^* = F(a_4, t_4)\&F(ack_4, t_4)\&H(t_4, a_5)\&P(a_4)\&P(ack_4) \to \neg P(a_4)\&\neg P(ack_4)\&P(a_5)$;

u_4: $mu_4^* = F(b_4, u_4)\&H(u_4, ack_4)\&H(u_4, b_5)\&P(b_4) \to \neg P(b_4)\&P(ack_4)\&P(b_5)$;

t_5: $mt_5^* = F(a_5, t_5)\&H(t_5, a_1)\&P(a_5) \to \neg P(a_5)\&P(a_1)$;

u_5: $mu_5^* = F(b_5, u_5)\&H(u_5, b_1)\&P(b_5) \to \neg P(b_5)\&P(b_1)$.

Depending on the structural links established in the network of RPN between positions and transitions, one or another implementation of the network is established. In this case, for each RPN transition, two fire rules are defined and, with the established structure, for each transition, only one rule is chosen for possible execution.

For the network shown in figure 1 and figure 2, an example of the RPN network reconfiguration will be given, which makes it possible to switch from the network in figure 1 to the network in figure 2, that is, when the reconfiguration, the transition from an exchange of the type "Client → Server" to the exchange of the type "Server → Client".

1. All links of the positions mes_1, ack_1, mes_2, ack_2 with the transitions (figure 1) are broken:

$H(t_1, mes_1) \leftarrow false$; $F(mes_1, u_1) \leftarrow false$; $H(u_2, ack_1) \leftarrow false$;
$F(ack_1, t_2) \leftarrow false$; $H(u_3, mes_2) \leftarrow false$; $F(mes_2, t_3) \leftarrow false$;
$H(t_4, ack_2) \leftarrow false$; $F(ack_2, u_4) \leftarrow false$.

(It is assumed that these statements were true before the operations were performed).

2. New links are established for the positions mes_3, ack_3, mes_4, ack_4 with transitions (figure 2):

$H(u_1, mes_3) \leftarrow true$; $F(mes_3, t_1) \leftarrow true$; $H(t_2, ack_3) \leftarrow true$;
$F(ack_3, u_2) \leftarrow true$; $H(t_3, mes_4) \leftarrow true$; $F(mes_4, u_3) \leftarrow true$;
$H(u_4, ack_4) \leftarrow true$; $F(ack_4, t_4) \leftarrow true$.

(It is assumed that these statements were false before the operations were performed).

The advantage of this approach to the construction of RPN networks is that the structural links between positions and transitions are specified by propositional functions, or predicates, F and H, which substantially simplifies the parametric adjustment of the network to the required mode of operation.

6. Simplification of production rules

The process of reconfiguring networks of the RPN type may in many cases not involve all structural links between positions and transitions, therefore, some of the true statements of the form $F(p_i, t_i)$ and $H(t_m, p_n)$ involved in the conjunctions can be simply omitted. Then the system of simplified production rules R_1', obtained by excluding constantly true statements from the system R_1, will take the following form:

The simplified production rules system R_1':

t_1: $mt_1 = H(t_1, mes_1)\&P(a_1) \to \neg P(a_1)\&P(a_2)\&P(mes_1)$;
u_1: $mu_1 = F(mes_1, u_1)\&P(mes_1)\&P(b_1) \to \neg P(mes_1)\& \neg P(b_1)\&P(b_2)$;

t_2: $mt_2 = F(ack_1, t_2)\&P(a_2)\&P(ack_1) \to \neg P(a_2)\&\neg P(ack_1)\&P(a_3)$;

u_2: $mu_2 = H(u_2, ack_1)\&P(b_2) \to \neg P(b_2)\&P(ack_1)\&P(b_3)$;

t_3: $mt_3 = F(mes_2, t_3)\&P(a_3)\&P(mes_2) \to \neg P(a_3)\&\neg P(mes_2)\&P(a_4)$;

u_3: $mu_3 = H(u_3, mes_2)\&P(b_3) \to \neg P(b_3)\&P(mes_2)\&P(b_4)$;

t_4: $mt_4 = H(t_4, ack_2)\&P(a_4) \to \neg P(a_4)\&P(a_5)\&P(ack_2)$;

u_4: $mu_4 = F(ack_2, u_4)\&P(b_4)\&P(ack_2) \to \neg P(b_4)\&\neg P(ack_2)\&P(b_5)$;

t_5: $mt_5 = P(a_5) \to \neg P(a_5)\&P(a_1)$;

u_5: $mu_5 = P(b_5) \to \neg P(b_5)\&P(b_1)$.

Similarly, based on R_2, a system of simplified product rules R_2' is formed:

The simplified production rules system R_2':

t_1: $mt_1^* = F(mes_3, t_1)\&P(a_1)\&P(mes_3) \to \neg P\ (a_1)\&\neg P\ (mes_3)\&P(a_2)$;

u_1: $mu_1^* = H(u_1, mes_3)\&P(b_1) \to \neg P\ (b_1)\&P(mes_3)\&P(b_2)$;

t_2: $mt_2^* = H(t_3, ack_3)\&P(a_2) \to \neg P\ (a_2)\&P(a_3)\&P(ack_3)$;

u_2: $mu_2^* = F(ack_3, u_2)\&P(ack_3)\&P(b_2) \to \neg P(ack_3)\&\neg P\ (b_2)\&P\ (b_3)$;

t_3: $mt_3^* = H(t_3, mes_4)\&P(a_3) \to \neg P\ (a_3)\&P(a_4)\&P(mes_4)$;

u_3: $mu_3^* = F(mes_4, u_3)\&P(mes_4)\&P(b_3) \to \neg P(mes_4)\&\neg P(b_3)\&P(b_4)$;

t_4: $mt_4^* = F(ack_4, t_4)\&P(a_4)\&P(ack_4) \to \neg P(a_4)\&\neg P(ack_4)\&P(a_5)$;

u_4: $mu_4^* = H(u_4, ack_4)\&P(b_4) \to \neg P\ (b_4)\&P(ack_4)\&P(b_5)$;

t_5: $mt_5^* = P(a_5) \to \neg P(a_5)\&P(a_1)$;

u_5: $mu_5^* = P(b_5) \to \neg P\ (b_5)\&P(b_1)$.

7. Logical-algebraic operational expressions for reconfigurable Petri nets

Direct programming of distributed applications with the help of production expressions is difficult, because of the operational semantics of product rules is not obvious. Therefore, for an explicit representation of the procedural and operational semantics of models originally represented by RPN networks, it is advisable to move from the system of production rules to logical-algebraic operational expressions (LAOE) describing the functioning of the RPN. Therefore, the transition happens from the systems of product rules R_1 and R_2 to LAOEs L_1 and L_2, respectively.

The expressions for the LAOEs form are constructed on the basis of the superposition principle of α-disjunctions introduced by V. M. Glushkov in systems of algorithmic algebras.

The α-disjunction operation allows to represent the operator *if*: $[\alpha](A_1 \vee A_2)$ in an analytic form. Here, depending on the true or false value of the condition α, the operator A_1 or the operator A_2, respectively, is executed. Further, the superposition of two operations of α-disjunction is also used: $[\alpha_1]([\alpha_2](A_1 \vee A_2) \vee A_3)$, where among the operators there can be an empty or identical operator E. In addition, the modification predicate P operators are used, for example, $P(a_1)\leftarrow$**false**, $P(a_2)\leftarrow$**true**. This presentation has clear operational semantics and allows the creation of modular network software.

The system of logical-algebraic operational expressions L_1:

t_1: $Lmt_1 = [F(a_1, t_1)\&H(t_1, a_2)\&H(t_1, mes_1)]([P(a_1)](\{P(a_1)\leftarrow$**false**, $P(a_2)\leftarrow$**true**, $P(mes_1)\leftarrow$**true**$\} \vee E) \vee E)$;

u_1: $Lmu_1 = [F(b_1, u_1)\&F(mes_1, u_1)\&H(u_1, b_2)]$

$([P(mes_1)\&P(b_1)](\{P(mes_1)\leftarrow$**false**, $P(b_1)\leftarrow$**false**, $P(b_2)\leftarrow$**true**$\} \vee E) \vee E)$;

t_2: $Lmt_2 = [F(a_2, t_2)\&F(ack_1, t_2)\&H(t_2, a_3)]([P(a_2)\&P(ack_1)](\{P(a_2)\leftarrow$**false**, $P(ack_1)\leftarrow$**false**, $P(a_3)\leftarrow$**true**$\} \vee E) \vee E)$;

u_2: $Lmu_2 = [F(b_2, u_2)\&H(u_2, ack_1)\&H(u_2, b_3)]([P(b_2)](\{P(b_2)\leftarrow$**false**, $P(ack_1)\leftarrow$**true**, $P(b_3)$ $\leftarrow$ **true**$\} \vee E) \vee E)$;

t_3: $Lmt_3 = [F(a_3, t_3)\&F(mes_2, t_3)\&H(t_3, a_4)]([P(a_3)\&P(mes_2)](\{P(a_3)\leftarrow$**false**, $P(mes_2)\leftarrow$**false**, $P(a_4)\leftarrow$**true**$\} \vee E) \vee E)$;

u_3: $Lmu_3 = [F(b_3, u_3)\&H(u_3, mes_2)\&H(u_3, b_4)]([P(b_3)](\{P(b_3)\leftarrow$**false**,

$P(mes_2)\leftarrow$ **true**, $P(b_4)\leftarrow$**true**$\} \vee E) \vee E)$;

t_4: $Lmt_4 = [F(a_4, t_4)\&H(t_4, a_5)\&H(t_4, ack_2)]([P(a_4)](\{P(a_4)\leftarrow$**false**, $P(a_5)\leftarrow$**true**, $P(ack_2)\leftarrow$ **true**$\} \vee E) \vee E)$;

u_4: $Lmu_4 = [F(ack_2, u_4)\&F(b_4, u_4)\&H(u_4, b_5)]([P(b_4)\&P(ack_2)](\{P(b_4)\leftarrow$**false**, $P(ack_2)\leftarrow$ **false**, $P(b_5)\leftarrow$**true**$\} \vee E) \vee E)$;

t_5: $Lmt_5 = [F(a_5, t_5)\&H(t_5, a_1)]([P(a_5)](\{P(a_5)\leftarrow$**false**, $P(a_1)\leftarrow$**true**$\} \vee E) \vee E)$;

u_5: $Lmu_5 = [F(b_5, u_5)\&H(u_5, b_1)]([P(b_5)](\{P(b_5)\leftarrow$**false**,

$P(b_1)\leftarrow$**true**$\} \vee E) \vee E)$.

The system of logical-algebraic operational expressions L_2:

t_1: $Lmt_1^* = [F(a_1, t_1)\&F(mes_3, t_1)\&H(t_1, a_2)]([P(a_1)\&P(mes_3)](\{P(a_1)\leftarrow$**false**,

$P(mes_3)\leftarrow$**false**, $P(a_2)\leftarrow$**true**$\} \vee E) \vee E)$;

u_1: $Lmu_1^* = [F(b_1, u_1)\&H(u_1, mes_3)\&H(u_1, b_2)]([P(b_1)](\{P(b_1)\leftarrow$**false**, $P(mes_3)\leftarrow$**true**, $P(b_2)\leftarrow$**true**$\} \vee E) \vee E)$;

t_2: $Lmt_2^* = [F(a_2, t_2)\&H(t_2, a_3)\&H(t_3, ack_3)]([P(a_2)](\{P(a_2)\leftarrow$**false**, $P(a_3)\leftarrow$**true**, $P(ack_3)\leftarrow$**true**$\} \vee E) \vee E)$;

u_2: $Lmu_2^* = [F(b_2, u_2)\&F(ack_3, u_2)\&H(u_2, b_3)]$

$([P(ack_3)\&P(b_2)](\{P(ack_3)\leftarrow$**false**, $P(b_2)\leftarrow$**false**, $P(b_3)\leftarrow$**true**$\} \vee E) \vee E)$;

t_3: $Lmt_3^* = [F(a_3, t_3)\&H(t_3, a_4)\&H(t_3, mes_4)]([P(a_3)](\{P(a_3)\leftarrow$**false**, $P(a_4)\leftarrow$**true**, $P(mes_4)\leftarrow$**true**$\} \vee E) \vee E)$;

u_3: $Lmu_3^* = [F(b_3, u_3)\&F(mes_4, u_3)\&H(u_3, b_4)]$

$([P(mes_4)\&P(b_3)](\{P(mes_4)\leftarrow$**false**, $P(b_3)\leftarrow$**false**, $P(b_4)\leftarrow$**true**$\} \vee E) \vee E)$;

t_4: $Lmt_4^* = [F(a_4, t_4)\&F(ack_4, t_4)\&H(t_4, a_5)]([P(a_4)\&P(ack_4)](\{P(a_4)\leftarrow$**false**, $P(ack_4)\leftarrow$**false**, $P(a_5)\leftarrow$**true**$\} \vee E) \vee E)$;

u_4: $Lmu_4^* = [F(b_4, u_4)\&H(u_4, ack_4)\&H(u_4, b_5)]([P(b_4)](\{P(b_4)\leftarrow$**false**, $P(ack_4)\leftarrow$**true**, $P(b_5)\leftarrow$**true**$\} \vee E) \vee E)$;

t_5: $Lmt_5^* = [F(a_5, t_5)\&H(t_5, a_1)]([P(a_5)](\{P(a_5)\leftarrow$**false**,

$P(a_1)\leftarrow$**true**$\} \vee E) \vee E)$;

u_5: $Lmu_5^* = [F(b_5, u_5)\&H(u_5, b_1)]([P(b_5)](\{P(b_5)\leftarrow$**false**,

$P(b_1)\leftarrow$**true**$\}\vee E) \vee E)$.

For the given structural relationship between the transitions and positions that remain unchanged in the reconfiguration of the CPR, the systems of expressions L_1 and L_2 were simplified. As a result, expression systems L_1' and L_2' were obtained:

The system of reduced logical-algebraic operating expressions L_1':

t_1: $Lmt_1 = [H(t_1, mes_1)]([P(a_1)](\{P(a_1)\leftarrow$**false**, $P(a_2)\leftarrow$**true**,
$P(mes_1)\leftarrow$**true**$\} \vee E) \vee E)$;

u_1: $Lmu_1 = [F(mes_1, u_1)]([P(mes_1)\&P(b_1)](\{P(mes_1)\leftarrow$**false**, $P(b_1)\leftarrow$**false**, $P(b_2)\leftarrow$**true**$\} \vee \vee E) \vee E)$;

t_2: $Lmt_2 = [F(ack_1, t_2)]([P(a_2)\&P(ack_1)](\{P(a_2)\leftarrow$**false**, $P(ack_1)\leftarrow$**false**, $P(a_3)\leftarrow$**true**$\} \vee E) \vee E)$;

u_2: $Lmu_2 = [H(u_2, ack_1)]([P(b_2)](\{P(b_2)\leftarrow$**false**, $(ack_1)\leftarrow$**true**,

$P(b_3) \leftarrow$ **true**$\} \vee E) \vee E)$;

t_3: $Lmt_3 = [F(mes_2, t_3)]([P(a_3)\&P(mes_2)](\{P(a_3)\leftarrow$**false**, $P(mes_2)\leftarrow$**false**, $P(a_4)\leftarrow$**true**$\} \vee E) \vee E)$;

u_3: $Lmu_3 = [H(u_3, mes_2)]([P(b_3)](\{P(b_3)\leftarrow$**false**, $P(mes_2)\leftarrow$ **true**,

$P(b_4)\leftarrow$**true**$\} \vee E) \vee E)$;

t_4: $Lmt_4 = [H(t_4, ack_2)]([P(a_4)](\{P(a_4)\leftarrow$**false**, $P(a_5)\leftarrow$**true**,

$P(ack_2) \leftarrow$ **true**$\} \vee E) \vee E)$;

u_4: $Lmu_4 = [F(ack_2, u_4)]([P(b_4)\&P(ack_2)](\{P(b_4)\leftarrow$**false**, $P(ack_2)\leftarrow$ **false**, $P(b_5)\leftarrow$**true**$\} \vee \vee E) \vee E)$;

t_5: $Lmt_5 = [P(a_5)](\{P(a_5)\leftarrow$**false**, $P(a_1)\leftarrow$**true**$\} \vee E)$;

u_5: $Lmu_5 = [P(b_5)](\{P(b_5)\leftarrow$**false**, $P(b_1)\leftarrow$**true**$\} \vee E)$.

The system of reduced logical-algebraic operating expressions L_2':

t_1: $Lmt_1^* = [F(mes_3, t_1)]([P(a_1)\&P(mes_3)](\{P(a_1)\leftarrow$**false**, $P(mes_3)\leftarrow$**false**, $P(a_2)\leftarrow$**true**$\} \vee \vee E) \vee E)$;

u_1: $Lmu_1^* = [H(u_1, mes_3)]([P(b_1)](\{P(b_1)\leftarrow$**false**, $P(mes_3)\leftarrow$**true**,

$P(b_2) \leftarrow$**true**$\} \vee E) \vee E)$;

t_2: $Lmt_2^* = [H(t_3, ack_3)]([P(a_2)](\{P(a_2)\leftarrow$**false**, $P(a_3)\leftarrow$**true**,

$P(ack_3)\leftarrow$**true**$\} \vee E) \vee E)$;

u_2: $Lmu_2^* = [F(ack_3, u_2)]([P(ack_3)\&P(b_2)](\{P(ack_3)\leftarrow$**false**, $P(b_2)\leftarrow$**false**, $P(b_3)\leftarrow$**true**$\} \vee \vee E) \vee E)$;

t_3: $Lmt_3^* = [H(t_3, mes_4)]([P(a_3)](\{P(a_3)\leftarrow$**false**, $P(a_4)\leftarrow$**true**,

$P(mes_4) \leftarrow$**true**$\} \vee E) \vee E)$;

u_3: $Lmu_3^* = [F(mes_4, u_3)]([P(mes_4)\&P(b_3)](\{P(mes_4)\leftarrow$**false**, $P(b_3)\leftarrow$**false**, $P(b_4)\leftarrow$**true**$\} \vee \vee E) \vee E)$;

t_4: $Lmt_4^* = [F(ack_4, t_4)]([P(a_4)\&P(ack_4)](\{P(a_4)\leftarrow$**false**, $P(ack_4)\leftarrow$**false**, $P(a_5)\leftarrow$**true**$\} \vee E) \vee E)$;

u_4: $Lmu_4^* = [H(u_4, ack_4)]([P(b_4)](\{P(b_4)\leftarrow$**false**, $P(ack_4)\leftarrow$**true**,

$P(b_5)\leftarrow$**true**$\} \vee E) \vee E)$;

t_5: $Lmt_5^* = [P(a_5)](\{P(a_5)\leftarrow$**false**, $P(a_1)\leftarrow$**true**$\} \vee E)$;

u_5: $Lmu_5^* = [P(b_5)](\{P(b_5)\leftarrow$**false**, $P(b_1)\leftarrow$**true**$\} \vee E)$.

8. External control of the reconfigurable Petri nets

Considering the following way of managing the reconfiguration of the RPN, based on external management of the network by another, control, or supervisory network. In the simplest case, this control may be organized by using only one position (figure 5). To select one of the two

firing rules for each transition, the *Contr* entry is entered, which is connected with the transitions of the reconfigurable network by inhibitory (with non-blackened circles at the ends) or information arcs (with black circles at the ends). The inhibitor link of the *Contr* position with some transition blocks its operation if there is a label in it, and the information link blocks the transition when there is no label in the *Contr* position.

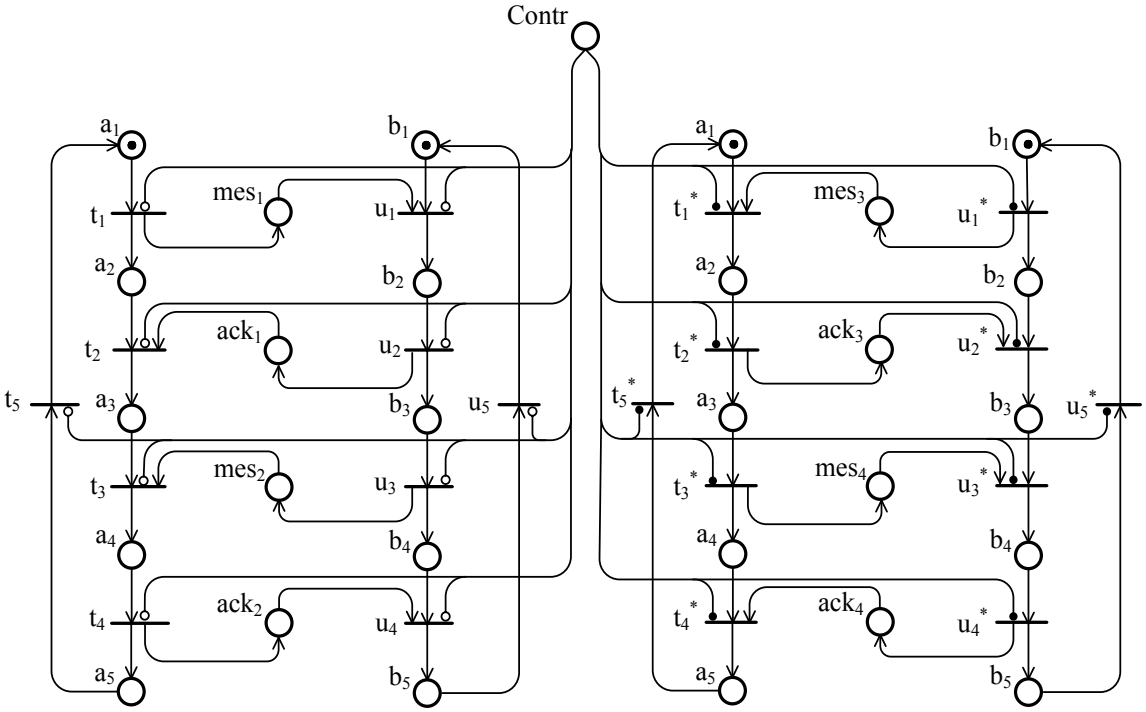


Fig. 5. Reconfigurable Petri net with external control from the Contr position of the supervisory network

Depending on the type of these links, one of the two firing rules is selected for each transition. The rules for firing transitions are presented below in the form of logical-algebraic operational expressions. Here $P(Contr)$ is a statement that is true when there is a marker in the control position *Contr*, *P* is an unary predicate markup symbol for the RPN network positions.

***The system of logical-algebraic operating expressions C_1 (external control of the RPN)*:**

t_1: $Cmt_1 = [\neg P(Contr)]([P(a_1)](\{P(a_1)\leftarrow$**false**, $P(a_2)\leftarrow$**true**,
$P(mes_1)\leftarrow$**true**$\} \vee E) \vee E)$;

u_1: $Cmu_1 = [\neg P(Contr)]([P(mes_1)\&P(b_1)](\{P(mes_1)\leftarrow$**false**, $P(b_1)\leftarrow$**false**, $P(b_2)\leftarrow$**true**$\} \vee E) \vee E)$;

t_2: $Cmt_2 = [\neg P(Contr)]([P(a_2)\&P(ack_1)](\{P(a_2)\leftarrow$**false**, $P(ack_1)\leftarrow$**false**, $P(a_3)\leftarrow$**true**$\} \vee E) \vee E)$;

u_2: $Cmu_2 = [\neg P(Contr)]([P(b_2)](\{P(b_2)\leftarrow$**false**, $(ack_1)\leftarrow$**true**,
$P(b_3) \leftarrow$ **true**$\} \vee E) \vee E)$;

t_3: $Cmt_3 = [\neg P(Contr)]([P(a_3)\&P(mes_2)](\{P(a_3)\leftarrow$**false**, $P(mes_2)\leftarrow$**false**, $P(a_4)\leftarrow$**true**$\} \vee \vee E) \vee E)$;

u_3: $Cmu_3 = [\neg P(Contr)]([P(b_3)](\{P(b_3)\leftarrow$**false**, $P(mes_2)\leftarrow$ **true**,
$P(b_4)\leftarrow$**true**$\} \vee E) \vee E)$;

t_4: $Cmt_4 = [\neg P(Contr)]([P(a_4)](\{P(a_4)\leftarrow$**false**, $P(a_5)\leftarrow$**true**,
$P(ack_2) \leftarrow$ **true**$\} \vee E) \vee E)$;

u_4: $Cmu_4 = [\neg P(Contr)]([P(b_4)\&P(ack_2)](\{P(b_4)\leftarrow$**false**, $P(ack_2)\leftarrow$ **false**, $P(b_5)\leftarrow$**true**$\} \vee \vee E) \vee E)$;

t_5: $Cmt_5 = [\neg P(Contr)]([P(a_5)](\{P(a_5)\leftarrow$**false**, $P(a_1)\leftarrow$**true**$\} \vee E) \vee E)$;

u_5: $Cmu_5 = [\neg P(Contr)]([P(b_5)](\{P(b_5)\leftarrow$**false**, $P(b_1)\leftarrow$**true**$\} \vee E) \vee E)$.

***The system of logical-algebraic operating expressions C_2 (external control of the RPN)*:**

t_1: $Cmt_1^* = [P(Contr)]([P(a_1)\&P(mes_3)](\{P(a_1)\leftarrow$**false**, $P(mes_3)\leftarrow$**false**,
$P(a_2)\leftarrow$**true**$\} \vee E) \vee E)$;

u_1: $Cmu_1^* = [P(Contr)]([P(b_1)](\{P(b_1)\leftarrow$**false**, $P(mes_3)\leftarrow$**true**,
$P(b_2) \leftarrow$**true**$\} \vee E) \vee E)$;

t_2: $Cmt_2^* = [P(Contr)]([P(a_2)](\{P(a_2)\leftarrow$**false**, $P(a_3)\leftarrow$**true**,
$P(ack_3)\leftarrow$**true**$\} \vee E) \vee E)$;

u_2: $Cmu_2^* = [P(Contr)]([P(ack_3)\&P(b_2)](\{P(ack_3)\leftarrow$**false**, $P(b_2)\leftarrow$**false**, $P(b_3)\leftarrow$**true**$\} \vee \vee E) \vee E)$;

t_3: $Cmt_3^* = [P(Contr)]([P(a_3)](\{P(a_3)\leftarrow$**false**, $P(a_4)\leftarrow$**true**,
$P(mes_4) \leftarrow$**true**$\} \vee E) \vee E)$;

u_3: $Cmu_3^* = [P(Contr)]([P(mes_4)\&P(b_3)](\{P(mes_4)\leftarrow$**false**, $P(b_3)\leftarrow$**false**, $P(b_4)\leftarrow$**true**$\} \vee \vee E) \vee E)$;

t_4: $Cmt_4^* = [P(Contr)]([P(a_4)\&P(ack_4)](\{P(a_4)\leftarrow$**false**, $P(ack_4)\leftarrow$**false**, $P(a_5)\leftarrow$**true**$\} \vee E) \vee E)$;

u_4: $Cmu_4^* = [P(Contr)]([P(b_4)](\{P(b_4)\leftarrow$**false**, $P(ack_4)\leftarrow$**true**,
$P(b_5)\leftarrow$**true**$\} \vee E) \vee E)$;

t_5: $Cmt_5^* =[P(Contr)]([P(a_5)](\{P(a_5)\leftarrow$**false**, $P(a_1)\leftarrow$**true**$\} \vee E) \vee E)$;

u_5: $Cmu_5^* =[P(Contr)]([P(b_5)](\{P(b_5)\leftarrow$**false**, $P(b_1)\leftarrow$**true**$\} \vee E) \vee E)$.

To explain the choice of the rule for firing a transition, for example, t_1. For this transition, one of the following two firing rules can be selected:

$Cmt_1 = [\neg P(Contr)]([P(a_1)](\{P(a_1)\leftarrow$**false**, $P(a_2)\leftarrow$**true**,
$P(mes_1)\leftarrow$**true**$\} \vee E) \vee E)$
or
$Cmt_1^* = [P(Contr)]([P(a_1)\&P(mes_3)](\{P(a_1)\leftarrow$**false**, $P(mes_3)\leftarrow$**false**,
$P(a_2)\leftarrow$**true**$\} \vee E) \vee E)$.

If there is no marker in the *Contr* position, then the statement $\neg P(Contr)$ is true and further, if the statement $P(a_1)$ is true, the actions in the curly brackets $\{P(a_1)\leftarrow$**false**, $P(a_2)\leftarrow$**true**, $P(mes_1)\leftarrow$**true**$\}$ are performed, modeling the removal of a marker from position a_1 and the appearance of markers in positions a_2 and mes_1, that is, the transition t_1 is firing in accordance with the rule Cmt_1.

If there is a marker in the *Contr* position, the statement $P(Contr)$ is true and further, if the statements $P(a_1)$ and $P(mes_3)$ are true, the actions in the curly brackets $\{P(a_1)\leftarrow$**false**, $P(mes_3)\leftarrow$**false**, $P(a_2)\leftarrow$**true**$\}$, modeling the removal of markers from positions a_1 and mes_3 and the appearance of a marker in position a_2, that is, the transition t_1 is firing in accordance with the rule Cmt_1^*. Similarly, the firing rules for the other transitions are selected.

Conclusion and future work

In RPN networks, if necessary, more complex management for the reconfiguration of the structure can be organized by using several external positions. The construction of the proposed kind of Petri nets is also based on a structural and reconfigurable approach, differing, however, taking into account the structural links and supervisory control. In addition, dynamic management for the reconfiguration of RPN can be organized, for example, after certain transitions are fired. For this, it will be necessary to define the links of these transitions with the supervisory positions. In this paper, the examples were used to construct a special type of Petri nets – RPNs, whose graphical representation contains duplicate positions. The work of such a Petri net is typical for peer-to-peer and other computer systems.

References

1. Peterson, J. L. Petri Net Theory and the Modeling of Systems / J. L. Peterson. – N. Y. : Prentice-Hall, 1981. – 288 p.

2. Murata, T. Petri nets: properties, analysis and applications / T. Murata // Proceedings of the IEEE. – 1989. – № 77 (4). – P. 541–580.

3. W. M. P. van der Aalst. A class of Petri nets for modeling and analyzing business processes // Computing Science Reports 95/26. – Eindhoven University of Technology, Eindhoven, 1995. – P. 1–25.

4. Hruz, B. Modeling and Control of Discrete-event Dynamic Systems with Petri Nets and Other Tools. Springer-Verlag / B. Hruz, M. C. Zhou. – London, 2007. – 351 p.

5. Iordache, M. V. Supervisory Control of Concurrent Systems. A Petri Net Structural Approach / M. V. Iordache, P. J. Antsaklis. – Boston : Birkhauser, 2006. – 281 p.

6. Jahns, R. On Modification in Petri Nets / R. Jahns, M. Becker, T. Bessey, H. Szczerbicka // Journal Proc. Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS). – 2002. – 5 p.

7. Valk, R. Self-modifying Nets / R. Valk // Report Bericht Nr. 34, IFI-HH-B-34/77, from the Institute of Informatics. – University of Hamburg, West Germany, 1977. – July. – 36 p.

8. Jiang, Z. Temporized Colored Petri Nets with Changeable Structure (TCPN-CS) for the Performance Modeling of Dynamic Production Systems / Z. Jiang, M. J. Zuo, R. Y. K. Fung, P. Y. Tu. // International Journal of Production Research. – 2000. – № 8. – P. 1917–1945.

9. Holloway, L. E. Survey of Petri Net Methods for Controlled Discrete Event Systems / L. E. Holloway, B. H. Krogh, A. A. Giua // Discrete Event Dynamic Systems: Theory and Applications. – Kluwer Academic Publishers, Boston : Manufactured in The Netherlands, 1997. – P. 151–190.

10. Krogh, B. H. Controlled Petri nets and maximally permissive feedback logic / B. H. Krogh // Proc. 25th Annual Allerton Conference. – University of Illinois, Urbana, 1987. – P. 317–326.

11. Ichikawa, A. Analysis and control of discrete event systems represented by Petri nets / A. Ichikawa, K. Hiraishi // Discrete Event Systems: Models and Applications, Lecture Notes in Control and Information Sciences / Varaiya, P. and Kurzhanski A. B., eds. – New York : Springer Verlag, 1988. – Vol. 103. – P. 115–134.

12. Omar, Yaqub. Modeling and Analysis of Connected Traffic Intersections Based on Modified Binary Petri Nets / Yaqub Omar, Li. Lingxi // Hindawi Publishing Corporation, International Journal of Vehicular Technology. – 2013. – 10 p. – URL: http://dx.doi.org/10.1155/2013/

13. Jianxin, L An Inter-Working Petri Net Model between SIMPLE and IMPS for XDM Service / Liao Jianxin, Yuting Zhang, Xiaomin Zhu // Petri Net, Theory and Applications. – 2008. – P. 73–90. – URL: www.intechopen.com

14. Petri Net Transformations / H. Ehrig, K. Hoffmann, J. Padberg, C. Ermel, U. Prange, E. Biermann, T. Modica // Chapter from the book Petri Net, Theory and Applications. – 2008. – P. 1–16. – URL: http://www.intechopen.com/books/petri_net_theory_and_applications

15. Petri Nets Fundamental Models // Verification and Applications / ed.by Michel Diaz. – John Wiley & Sons, 2009. – 585 p.

16. Svadova, M. Matlab Toolbox for Petri Nets / M. Svadova, Z. Hanzalek. – URL: https://simonline.akademitelkom.ac.id/dosen/riset/smartgrid/Bukan/Hanzalek01a.pdf

17. Petri Nets Tools Database Quick Overview. – URL: www.informatik.uni-hamburg.de

Mustafa Sadeq Jaafar, Sergey A. Zinkin

Penza State University, Penza, Russia

THE IMPLEMENTATION OF GLOBAL COMPUTING THROUGH THE MAPPING OF OBJECT-ORIENTED PETRI NETS INTO THE ARCHITECTURE OF DISTRIBUTED COMPUTING SYSTEMS

The article deals with the problems of mapping Petri nets to the system and functional architectures of distributed computing systems at the middleware level. In particular, the problem of mapping Petri nets to the TCP/IP network is solved experimentally. The need to solve this problem came from the fact that various methods for mapping algorithms to the architecture of parallel and distributed computing systems are known, but only a few of them are oriented to the development of middleware. The importance of the problem is explained by the fact that the definition of middleware overlaps with many existing technologies, such as Cloud Computing, Grid Computing, Utility Computing, Services Computing, and Distributed Computing in general. It is shown that different interpretations of its work for the same Petri net can be used, which significantly affect the quality of the middleware class network software. An example and detailed comments for the distributed Java program are given.

1. Introduction

Cloud Computing is defined as a large-scale distributed computing paradigm that is driven by economies of scale, in which a pool of abstracted, virtualized, dynamically-scalable, managed computing power, storage, platforms, and services are delivered on demand to external customers over the Internet. This definition of Cloud Computing overlaps with many existing technologies, such as Grid Computing, Utility Computing, Services Computing, and Distributed Computing in general [1]. The problems in this area cover progressively the hardware and software architectures that underpin modern massively parallel computer systems; the concepts and technologies that enable cloud and distributed computing; and advanced topics in distributed computing, including grid, peer-to-peer, and the Internet of Things [2].

The main areas related to the development of Cloud Computing and Grid Computing cover the following topics [3]:

– optimal online deterministic algorithms and adaptive heuristics for energy and performance;

– efficient dynamic consolidation of virtual machines in cloud data centers;

– a check-pointing enabled and resource-aware Java Virtual Machine (VM) for efficient and reliable e-science applications in grid environments;

– a bi-criteria scheduling process with Class of Services support on Grids and Clouds;

– a component-based middleware for hybrid Grid/Cloud Computing platforms;

– a peer-to-peer scheduling mechanism for workflows in Grid Computing;

– a provenance management system for scientific workflows;

– an adaptive parallel execution strategy for cloud-based scientific workflows etc.

When building systems similar to this distributed “Cloud” or “Grid” systems, a convenient toolkit, based on formalized specifications of their system and functional architecture is needed. Data processing technologies based on distributed object-oriented Petri nets (DOOPN) allow building real distributed architectures. Despite the growing popularity of the use of formal tools based Petri nets, when creating prototypes of distributed computing systems on the hardware and

software platforms of local and global networks, the implementation of such tools has not yet found wide distribution, as indicated by an overview of known tools from the site http://www.informatik.uni-hamburg.de/TGI/PetriNets/tools/quick [4]. A Java-program is proposed to implement one of the variants of such DOOPN networks. It is also possible to implement this program in the form of an applet, but in this case serious limitations are imposed on the configuration of DOOPN, due to the fact that the applet can establish a connection on the network only with the node from which it was downloaded. The technology of the DOOPN networks is closely related to networked computer technologies.

2. Interpretation of object-oriented Petri nets for applications to computer networks

First of all, some known information on Petri nets were given from [5, 6]. Petri nets were introduced in 1962 by Dr. Carl Adam Petri [6]. Additional information on Petri nets can be found in [7, 8] and others. A further description of conventional Petri nets is based on the citations from [5]: "A Petri net is a particular kind of bipartite directed graphs populated by three types of objects. These objects are *places*, *transitions*, and *directed arcs*. Directed arcs connect places to transitions or transitions to places. In order to study the dynamic behavior of a Petri net modeled system in terms of its states and state changes, each place may potentially hold either none or a positive number of *tokens*. Tokens are a primitive concept for Petri nets in addition to places and transitions. The presence or absence of a token in a place can indicate whether a condition associated with this place is true or false, for instance.

A Petri net is formally defined as a 5-tuple $N = (P, T, F, H, M_0)$, where

(1) $P = \{p_1, p_2, \ldots, p_m\}$ is a finite set of places;

(2) $T = \{t_1, t_2, \ldots, t_n\}$ is a finite set of transitions, $P \cup T \neq \varnothing$, and $P \cap T = \varnothing$;

(3) F: $P \times T \rightarrow N$ is an *input function* that defines directed arcs from places to transitions, where N is a set of nonnegative integers;

(4) H: $T \times P \rightarrow N$ is an *output function* that defines directed arcs from transitions to places;

(5) M_0: $P \rightarrow N$ is the *initial marking*.

A *marking* in a Petri net is an assignment of tokens (or markers) to the places of a Petri net. Tokens reside in the places of a Petri net. The number and position of tokens may change during the execution of a Petri net. The tokens are used to define the execution of a Petri net.

The execution of a Petri net is controlled by the number and distribution of tokens in a Petri net. By changing the distribution of tokens in places, it may reflect the occurrence of events or execution of operations, for instance, it could help in studying the dynamic behavior of the modeled system. A Petri net executes by *firing* transitions. Now the enabling rule and firing rule of a transition, which govern the flow of tokens would be introduced in the following points:

(1) *Enabling Rule*: A transition t is said to be *enabled* if each input place p of t contains at least the number of tokens equals to the weight of the directed arc connecting p to t, i.e., $M(p) \geq F(p, t)$ for any p in P.

(2) *Firing Rule*: Only enabled transitions can be fired. The firing of an enabled transition t removes from each input place p the number of tokens equals to the weight of the directed arc connecting p to t. It also deposits in each output place p the number of tokens equals to the weight of the directed arc connecting t to p. Mathematically, firing t at M yields a new marking

$$M'(p) = M(p) - F(p, t) + H(t, p) \text{ for any } p \text{ in } P.$$

Notice that since only enabled transitions can fire, the number of tokens in each place always remains non-negative when a transition is fired. Firing transitions would never try to remove a token that is not there." On this, citing the description of the most famous Petri nets from [5] is finished.

Petri nets used in this work differ in that markers can have parameters in them, and input and output arcs can be weighted by conditions.

In this paper, a formalized method was proposed for creating network applications of the "middleware" class by embedding the Petri net into the architecture of computer networks for global computing. First, consider embedding to a computer network of a Petri net fragment containing one transition t_1, three inputs (p_1, p_2, p_3) and one output (p_4) positions. The rule r_1 for firing the transition t_1 is represented in the form of a logical-algebraic expression:

$$r_1 = [Mark(p_1)\&Mark(p_2)\&Mark(p_3)](\{Mark(p_1)\leftarrow\textbf{false}, Mark(p_2)\leftarrow\textbf{false}, Mark(p_3)\leftarrow\textbf{false}, Mark(p_4)\leftarrow\textbf{true}\} \vee E), \quad (1)$$

where $Mark$: $P \rightarrow \{\textbf{true}, \textbf{false}\}$ is an unary marking function defined for a 1-bounded fragment of the Petri net, $\alpha = Mark(p_1)\&Mark(p_2)\&Mark(p_3)$ – compound statement, the truth of which allows the firing of the transition t_1. Updating $Mark(p_i)\leftarrow\textbf{true}$ of the function $Mark$ means placing the marker in the position p_i, and updating $Mark(p_j)\leftarrow\textbf{false}$ of the same function – removing the marker from the position p_j. As soon as the condition α becomes true, the following updates to the function $Mark$ are performed: $\{Mark(p_1)\leftarrow\textbf{false}, Mark(p_2)\leftarrow\textbf{false}, Mark(p_3)\leftarrow\textbf{false}, Mark(p_4)\leftarrow\textbf{true}\}$. The curly brackets here have an important meaning: all the operations indicated in them are considered to be consistent. These operations can be executed in an arbitrary order or in parallel. The symbol E denotes an empty operator, which is satisfied in the case when the compound statement α is false.

First of all, it's necessary to consider the interpretation of Petri nets for applications to computer networks. The testing of the condition α can have a cyclic character, therefore, the operator E is replaced by the operator $Ret_{\tau,n}$:

$$r'_1 = [Mark(p_1)\&Mark(p_2)\&Mark(p_3)](\{Mark(p_1)\leftarrow\textbf{false}, Mark(p_2)\leftarrow\textbf{false}, Mark(p_3)\leftarrow\textbf{false}, Mark(p_4)\leftarrow\textbf{true}\} \vee Ret_{\tau,n}). \quad (2)$$

The operator $Ret_{\tau,n}$ returns control to check condition $\alpha = Mark(p_1)\&Mark(p_2)\&Mark(p_3)$ until it is true, or the number of re-attempts to fire the transition will be exhausted. Each repetition of an attempt to fire a transition t_1 is carried out after a certain time intervalτ.

In the general case, the expression

$$A;[\alpha](B \vee Ret_{\tau,n});C \quad (3)$$

corresponds to the following algorithm shown in figure 1. The symbol ";" here means the sequential execution of operators.

As it was stated above that the operations $\{Mark(p_1)\leftarrow\textbf{false}, Mark(p_2)\leftarrow\textbf{false}, Mark(p_3)\leftarrow\textbf{false}, Mark(p_4)\leftarrow\textbf{true}\}$ can be executed in an arbitrary order or in parallel one. However, during the interpretation of expressions (1) and (2) in computer networks and microprocessor systems, the order of operations is important. Consider, for example, the following expression:

$$r''_1 = [Mark(p_1)\&Mark(p_2)\&Mark(p_3)](\{(Mark(p_1)\leftarrow\textbf{false}, Mark(p_2)\leftarrow\textbf{false}, Mark(p_3)\leftarrow\textbf{false}); Mark(p_4)\leftarrow\textbf{true}\} \vee Ret_{\tau,n}). \quad (4)$$

In a sequence of statements enclosed in braces in (3), the symbols "," and ";" mean, respectively, arbitrary (possibly parallel) and sequential execution of actions.

Similarly, when implemented in computer networks, there can have a meaning of an order for verification of statements $Mark(p_1)$, $Mark(p_2)$ and $Mark(p_3)$, since this can be related to the order in which messages are sent and received in the computer network.

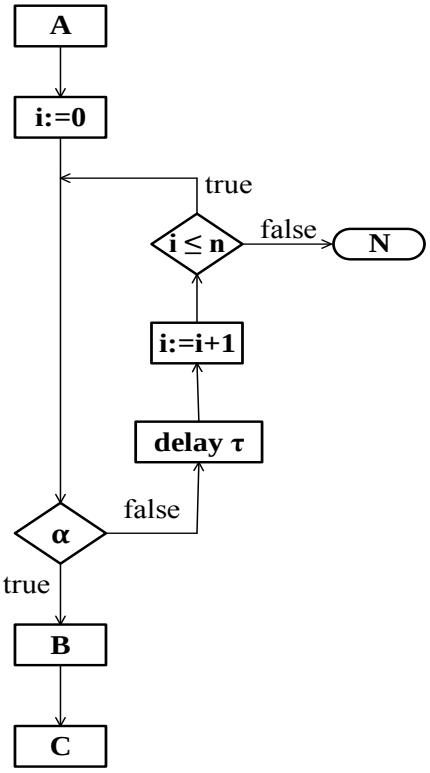


Fig. 1. Algorithm for the network implementation of the transition firing rule

Considering the case where the positions p_1, p_2, p_3, p_4 and the transition t_1 are placed on the nodes n_{p1}, n_{p2}, n_{p3}, n_{p4} and n_{t1} of the computer TCP/IP network, respectively (figure 2). Two unary functions f_{Node} and $f_{Address}$ were introduced:

$$f_{Node}: P \cup T \to Node;$$
$$f_{Address}: Node \to IPaddress,$$

where *Node* is a set of nodes that belongs to a computer network, *IPaddress* – a set of IP-addresses in a TCP/IP network. For example, the expression $f_{Address}(f_{Node}(p_1))$ allows you to determine the IP address 129.44.192.1 of the node $n_{p1} = f_{Node}(p_1)$ on which position p_1 is implemented.

Three variants for the implementation of the transition t_1 firing rule in a TCP/IP network were considered in this paper:

1. For the implementation of the rule r'_1 for firing the transition t_1, the node n_{t1} on which the transition t_1 is implemented makes a request for the markers to be present at the nodes n_{p1}, n_{p2} and n_{p3} and then initiates an independent and possibly a parallel transfer of their markers to the node n_{t1}. Then, the node n_{t1} sends a new marker to node n_{p4}, where the output position p_4 is implemented. The function updates $Mark(p_1) \leftarrow$ **false**, $Mark(p_2) \leftarrow$ **false** and $Mark(p_3) \leftarrow$ **false** correspond to sending acknowledgments to the node-positions that have been freed from the markers. The function update $Mark(p_4) \leftarrow$ **true** corresponds to the sending of the marker from the node-transition n_{t1} to the output node-position n_{p4}. The implementation of the rule r''_1 differs in that, the output marker is sent to the output node-position n_{p4} at the very final stage.

2. The second variant for the implementation of the firing rules is that, when polling, the markers in the input node-positions are blocked after their finding. Node-transition n_{t1} will then fire only if all markers will be present in the input positions (after receiving all of the markers-messages). Then, the output marker is sent from the node-transition n_{t1} to the output node-position n_{p4}.

3. In the third case, all the markers-messages from the input node-positions n_{p1}, n_{p2} and n_{p3} are immediately and independently transferred to the node-transition n_{t1}. Then, the output marker is sent from node-transition n_{t1} to the output node-position n_{p4}.

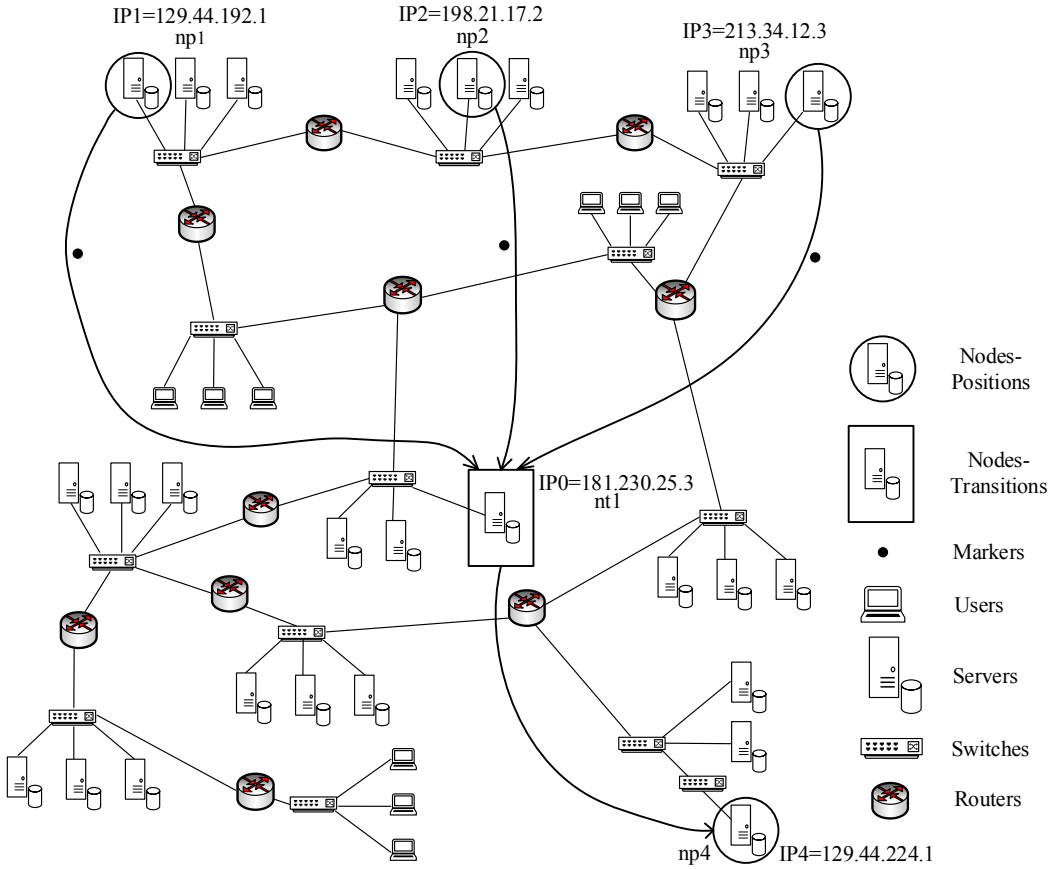


Fig. 2. An example of mapping object-oriented Petri nets into the architecture of distributed computing systems based on TCP/IP networks

Other implementations of the rules for firing transitions when Petri nets are embedded in the architecture of a computer network are also possible, so when developing middleware, the developer must choose an acceptable option.

3. The principles of creating a program for mapping object-oriented Petri nets into the architecture of distributed computing systems

Direct mapping of object-oriented Petri nets into the architecture of distributed computing systems is the most accessible to application software developers, as well as to developers of intermediate-type middleware software. A description of the demo version of the program, on the basis of which complex distributed applications can be created with a structure represented as a Petri net, is given below.

The object-oriented Petri net is represented by the classes Place and Transition. Between the objects of these classes, a marker which is implemented in the Mark class and transmitted over the TCP/IP network. To work with the TCP/IP network, the java.net package was used, in particular, the classes InetAddress, Socket and ServerSocket.

The Mark class describes a marker transmitted over the Petri net:

```
class Mark { …
```

Each marker can have parameters stored in the array parameter. The number of parameters in the markers is fixed:

```
final int MAX_PARAM = 3;
byte[] parametr = new byte[MAX_PARAM];
```

The marker constructor sets the parameter values:

```
Mark(){
    for (int i=0; i<MAX_PARAM; i++)
          parametr[i]=0;
    }
}
```

The Place class implements the network position:

```
class Place {
    private final int MAX_OUT = 1;
```

The out_adr and out_port arrays store information about arcs emerging from the position:

```
private InetAddress[] out_adr =  new netAddress[MAX_OUT];
    private int out_port[] = new int[MAX_OUT];
    private int in_port;
```

In this implementation, the position can contain only one marker:

```
    private Mark m;
```

The sendMark method dispatches markers from the position:

```
void sendMark() throws IOException{
          for(int i=0; i<MAX_OUT; i++){
               Socket s = new Socket(out_adr[i], out_port[i]);
            s.getOutputStream().write(m.parametr);
            s.close();
          }
    }
```

The position constructor creates a node with the given input and output links:

```
Place(int i_port, InetAddress[] o_adr, int[] o_port) {
          for(int i=0; i<MAX_OUT; i++){
              out_adr[i] = o_adr[i];
               out_port[i] = o_port[i];
          };
          in_port = i_port;
    }
}
```

The Transition class represents the transition of the Petri net:

```
class Transition {
    private final int MAX_OUT = 1;
    private final int COUNT_IN_MARK = 3;
```

The addresses of the output nodes of the network:

```
    private InetAddress[] out_adr = new InetAddress[MAX_OUT];
    private int out_port[] = new int[MAX_OUT];
    private int in_port;
```

The marker to be sent to the output nodes:

```
private Mark m_out_mark;
```

Markers needed to run the transition:

```
private Mark[] m_in_mark = new Mark[COUNT_IN_MARK];
    private static int n=0;
```

The transition constructor sets up the input and output links:

```
  Transition(int i_port, InetAddress[] o_adr, int[] o_port) {
        for(int i=0; i<MAX_OUT; i++){
            out_adr[i] = o_adr[i];
            out_port[i] = o_port[i];
        }
        in_port = i_port;
    }
```

The firing method contains the actions that are performed when the transition is started. If necessary, you can include more complex processes of marker parameters, as well as increase the number of outgoing markers for the implementation of conditional branching:

```
  void fire() {
        // transition's events
        m_out_mark = m_in_mark[0];
    }
```

The enable method checks for the presence of the tokens needed to start the transition:

```
boolean enable() {
        for(int i=0; i<COUNT_IN_MARK; i++)
            if (m_in_mark[i] == null)
                return false;
        return true;
    }
```

The sendMark method dispatches markers from the transition. Here, a conditional transfer of a marker can also be implemented:

```
void sendMark() throws IOException{
        int i;
        for(i=0; i<MAX_OUT; i++){
            Socket s = new Socket(out_adr[i], out_port[i]);
            s.getOutputStream().write(m_out_mark.parametr);
            s.close();
        }
        n=0;
        for(i=0; i<COUNT_IN_MARK; i++)
             m_in_mark[i] = null;
    }
```

The recvMark method accepts tokens:

```
void recvMark() throws IOException{
        ServerSocket ss = new ServerSocket(in_port);
        Socket s = ss.accept();
        InputStream ins = s.getInputStream();
        ins.read(m_in_mark[n++].parametr);
        s.close();
        ss.close();
    }
    void work() throws IOException{
        while (!enable())
            recvMark();
        fire();
        sendMark();
    }
}
```

The creation of a Petri net can be performed as follows, The PlaceNode class implements three node-positions that are connected to one transition:

```
class PlaceNode {
  public static void main(String[] args) throws Exception{
        // Positions creation
        Place place1, place2, place3;
        InetAddress[] outAdr = new InetAddress[1];
        int[] outPort = new int[1];
        outAdr[0] = InetAddress.getByName("127.0.0.1");
        outPort[0] = 12100;
```

```
            // Linking positions with a transition
            place1 = new Place(12001, outAdr, outPort);
            place2 = new Place(12002, outAdr, outPort);
            place3 = new Place(12003, outAdr, outPort);
            place1.sendMark();
            place2.sendMark();
            place3.sendMark();
            System.out.println(" All Marks sended.");
  }
}
```

The TransitionNode class implements a junction node with three input arcs and one output:

```
class TransitionNode {
  public static void main(String[] args) throws Exception{
            InetAddress[] outAdr = new InetAddress[1];
            int[] outPort = new int[1];
            outAdr[0] = InetAddress.getByName("127.0.0.1");
            outPort[0] = 12004;
            Transition tNode1 = new Transition(12100, outAdr, outPort);
            tNode1.work();
            System.out.println(" Transition fired.");
  }
}
```

An alternative way to implement the launching of both the PlaceNode and TransitionNode classes is to use lightweight processes. In this case, the specified classes are performed within the same process. This option is useful if several positions or transitions in the full text of the program of a Petri net fragment look like this, are implemented in one node of the network.

4. A program for mapping object-oriented Petri nets into the architecture of distributed computing systems

The full text of the program for the Petri net fragment of the DOOPN class looks like this:

```
import java.net.*;
import java.io.*;
class Mark {
      final int MAX_PARAM = 3;
      byte[] parametr = new byte[MAX_PARAM];

      Mark(){
            for (int i=0; i&ltMAX_PARAM; i++)
                  parametr[i]=0;
      }
}
```

```
class Place {
    private final int MAX_OUT = 1;
    private InetAddress[] out_adr = new InetAddress[MAX_OUT];
    private int out_port[] = new int[MAX_OUT];
    private int in_port;
    private Mark m;
    void sendMark() throws IOException{
        for(int i=0; i&ltMAX_OUT; i++){
            Socket s = new Socket(out_adr[i], out_port[i]);
            s.getOutputStream().write(m.parametr);
            s.close();
        }
    }
    Place(int i_port, InetAddress[] o_adr, int[] o_port) {
        for(int i=0; i&ltMAX_OUT; i++){
            out_adr[i] = o_adr[i];
            out_port[i] = o_port[i];
        };
        in_port = i_port;
    }
}
class Transition {
    private final int MAX_OUT = 1;
    private final int COUNT_IN_MARK = 3;
    private InetAddress[] out_adr = new InetAddress[MAX_OUT];
    private int out_port[] = new int[MAX_OUT];
    private int in_port;
    private Mark m_out_mark;
    private Mark[] m_in_mark = new Mark[COUNT_IN_MARK];
    private static int n=0;
    Transition(int i_port, InetAddress[] o_adr, int[] o_port) {
        for(int i=0; i&ltMAX_OUT; i++){
            out_adr[i] = o_adr[i];
            out_port[i] = o_port[i];
        }
        /*     for(int i=0; i&ltCOUNT_IN_MARK; i++)
            m_in_mark[i] = i_mark[i];*/
        in_port = i_port;
    }
    void fire() {
    // transition's events
        m_out_mark = m_in_mark[0];
    }
```

```
        boolean enable() {
            for(int i=0; i&ltCOUNT_IN_MARK; i++)
                if (m_in_mark[i] == null)
                    return false;
            return true;
        }
        void sendMark() throws IOException{
            int i;
            for(i=0; i&ltMAX_OUT; i++){
                Socket s = new Socket(out_adr[i], out_port[i]);
                s.getOutputStream().write(m_out_mark.parametr);
                s.close();
                n=0;
                for(i=0; i&ltCOUNT_IN_MARK; i++)
                    m_in_mark[i] = null;
            }
        }
        void recvMark() throws IOException{
            ServerSocket ss = new ServerSocket(in_port);
            Socket s = ss.accept();
            InputStream ins = s.getInputStream();
            ins.read(m_in_mark[n++].parametr);
            s.close();
            ss.close();
        }
        void work() throws IOException{
             while (!enable())
                  recvMark();
             fire();
             sendMark();
        }
}
class PlaceNode {
        public static void main(String[] args) throws Exception{
            Place place1, place2, place3;
            InetAddress[] outAdr = new InetAddress[1];
            int[] outPort = new int[1];
            outAdr[0] = InetAddress.getByName("127.0.0.1");
            outPort[0] = 12100;
            place1 = new Place(12001, outAdr, outPort);
            place2 = new Place(12002, outAdr, outPort);
            place3 = new Place(12003, outAdr, outPort);
            place1.sendMark();
```

```
            place2.sendMark();
            place3.sendMark();
            System.out.println(" All marks sended.");
        }
    }
    class TransitionNode {
        public static void main(String[] args) throws Exception{
            InetAddress[] outAdr = new InetAddress[1];
            int[] outPort = new int[1];
            outAdr[0] = InetAddress.getByName("127.0.0.1");
            outPort[0] = 12004;
            Transition tNode1 = new Transition(12100, outAdr, outPort);
            tNode1.work();
            System.out.println(" Transition fired.");
        }
    }
```

Conclusion

The given example is suitable for the realization in computer networks and demonstrates the mapping of a Petri net into the architecture of a distributed computing system. The realization of particular cases in some specific Petri networks of the DOOPN class requires the completion of this program, a specific interpretation of Petri net can take into account the priorities in triggering the transitions, the presence of additional conditions on the input and output arcs, the presence of inhibitory and information arcs, the multiplicity of arcs, the number of marks in positions, tag parameters and other properties.

References

1. Foster, Ian. Cloud Computing and Grid Computing 360-Degree Compared / Ian Foster, Yong Zhao, Ioan Raicu, Shiyong Lu. – URL: https://www.researchgate.net/publication/23716795_Cloud_Computing_and_Grid_Computing_360-Degree_Compared

2. Hwang, Kai. Distributed and Cloud Computing: From Parallel Processing to the Internet of Things / Kai Hwang, Geoffrey C. Fox, and Jack J. Dongarra. – Elsevier (Singapore) Pte Ltd, 2012. – 630 p.

3. Schulze, B. Middleware for Clouds and e-Science. Editorial / B. Schulze, R. Buyya, F. Porto // Concurrency Computation: Practice and Experience. – John Wiley & Sons, Ltd., 2012.

4. Petri Nets Tools Database Quick Overview. – URL: http://www.informatik.uni-hamburg.de/TGI/PetriNets/tools/quick

5. Wang, J. Petri Nets for Dynamic Event-Driven System Modeling / J. Wang. – Department of Software Engineering. Monmouth University, West Long Branch, NJ 07764. – URL: http://bluehawk.monmouth.edu

6. Petri, C. A. Kommunikation mit Automaten / C. A. Petri. – New York, 1962.

7. Peterson, J. L. Petri Net Theory and the Modeling of Systems / J. L. Peterson. – New York : Prentice-Hall, 1981. – 288 p.

8. Murata, T. Petri nets: properties, analysis and applications / T. Murata // Proceedings of the IEEE. – 1989. – № 77(4). – P. 541–580.

Г. В. Мартяшин, И. С. Пышкина, Д. А. Трокоз

Пензенский государственный университет, Пенза, Россия

ОСОБЕННОСТИ ПОСТРОЕНИЯ КАРТ ПОКРЫТИЯ В СИСТЕМАХ ПАССИВНОГО МОНИТОРИНГА ТЕХНИЧЕСКИХ ПАРАМЕТРОВ СЕТЕЙ СОТОВОЙ СВЯЗИ

Обсуждается вопрос пассивного мониторинга технических параметров сетей сотовой связи. Предлагается методология построения карт покрытия для систем, основанных на пассивном мониторинге.

Диагностика и мониторинг являются неотъемлемыми составляющими предоставления услуг сотовой связи. Это подразумевает выполнение измерений, которые осуществляются с помощью специализированных измерительных комплексов. В настоящее время сотовые операторы преимущественно используют измерительные комплексы, разработанные зарубежными производителями, которые отличаются высокой стоимостью и при этом требуют постоянного нахождения высококвалифицированного инженера непосредственно рядом с аппаратурой комплекса для управления процессом измерения [1]. Решение этой проблемы заключается в использовании пассивного способа мониторинга, который позволяет производить автоматизированный сбор данных и отказаться от целенаправленных измерений. Использование современного смартфона, обладающего возможностью определения местоположения с помощью GPS и доступом к сети интернет, в качестве основы измерительного комплекса легко позволяет реализовать пассивный мониторинг. Этим была продиктована необходимость разработки программного комплекса диагностики и мониторинга сетей сотовой связи на базе смартфона, использующего пассивный подход к мониторингу [2].

Сотовая связь имеет в своей основе ключевую особенность, которая состоит в том, что общая зона покрытия делится на части, каждая из которых называется сотой и обслуживается совей базовой станцией, обеспечивающей зону покрытия соты. В идеальных условиях ровной поверхности и отсутствия каких-либо помех зона покрытия базовой станции представляет собой круг, однако, так как с целью обеспечения непрерывного покрытия зоны покрытия соседних сот частично перекрывают друг на друга, соты традиционно рассматриваются как граничащие шестиугольные ячейки [3]. Таким образом, сеть сотовой связи представляет собой множество территориально разнесенных базовых станций, обеспечивающих непрерывную зону покрытия, и коммутационное оборудование, связывающее соты в единое целое с телефонными сетями общего пользования.

Результаты мониторинга технических параметров сетей сотовой связи заключаются в построении актуальной, полученной эмпирическим путем, карты покрытия, то есть карты, представляющей графическую проекцию сигнала базовых станций на местность с градацией уровня сигнала. Такая карта крайне интересна по той причине, что в силу виляния различных факторов, таких как особенности ландшафта, наличие построек, наличие источников достаточно сильных радиоволн, реальное покрытие может отличаться от теоретического. Помимо построения общей актуальной карты покрытия еще один немаловажным результатом мониторинга является построение локальных зон покрытия, соответствующих сотам. Подобная информация позволяет оценить соответствие реальной зоны покрытия, обеспечивающейся сотой, и теоретической зоны покрытия, что может поспособствовать обнаружению некорректности режима работы или расположения базовой станции.

Важным моментом является тот факт, что используемая для определения координат точки измерения система геопозиционирования GPS имеет погрешность, суммарное среднеквадратичное значение которой составляет 13,1 м [4]. Таким образом, под точкой измерения здесь понимается круглая область, радиус которой составляет 13,1 м, внутри которой находится реальная точка измерения.

Еще одним немаловажным замечанием является то, что все смартфоны, производящие измерения, в зависимости от марки могут иметь различные антенны и радиомодули, что непосредственно виляет на определение уровня принимаемого сигнала тем или иным смартфоном в одной и той же точке. Таким образом, значение измеренного уровня сигнала в точке измерения является случайной величиной, на которую влияют конструктивные особенности смартфона. Тогда, исходя из центральной предельной теоремы, которая утверждает, что сумма достаточно большого количества слабо зависимых случайных величин, имеющих примерно одинаковые масштабы (ни одно из слагаемых не доминирует, не вносит в сумму определяющего вклада), имеет распределение, близкое к нормальному, можно сделать вывод, что уровень сигнала в определенной точке, измеренный с помощью смартфона является случайной величиной, распределенной по нормальному закону [5]. Тогда выборочное среднее (выборкой в данном случае является совокупность результатов измерений) при точечной оценке математического ожидания нормально распределенного значения уровня сигнала можно считать состоятельной и несмещенной оценкой [6]. Итак, для оценки уровня сигнала в точке измерения, используется выборочное среднее, иначе говоря, среднее арифметическое совокупности результатов измерений уровня сигнала в этой точке.

При построении карт покрытия для наглядности градации уровня сигнала в той или иной области используются цвета, соответствующие ранговой оценке уровня сигнала в данной области.

Для того, чтобы по полученным данным построить карту покрытия с градацией уровня сигнала, предлагается использовать дискретную модель местности. Для этого на территорию, карта покрытия которой строится, накладывается сетка, разбивающая ее на одинаковые ячейки квадратной формы. Оптимальные размеры ячейки должны быть определены эмпирическим путем, однако, размеры ячейки должны быть соизмеримы с размерами области, в которой находится точка измерения, то есть круга с радиусом 13,1 м, для того, чтобы можно было сделать допущение о том, что в рамках квадрата уровень сигнала имеет примерно одинаковое значение, равное среднему значению уровней сигнала в точках, принадлежащих квадрату. В то же время размеры ячейки должен быть на порядок меньше размеров соты, чтобы возможно было отследить границы соты и изменение уровня сигнала внутри соты. Минимальный размер имеет сота сети типа UMTS (3G) в городских условия, для обеспечения оптимального режима работы ее радиус должен составлять в среднем 1–1,5 км. Исходя из описанных требований, размер дискретной ячейки карты покрытия определен как 30*30 м, однако, это предварительное допускаемое значение, которое, возможно, будет получено эмпирическим путем в дальнейшем для обеспечения эффективного построения карты покрытия.

Средний уровень сигнала, характеризующий ячейку, поддается ранговой оценке, в результате чего определяется соответствующий градации уровня сигнала цвет, в который на карте покрытия будет окрашена ячейка. Распределение точек измерения по дискретным ячейкам карты покрытия проиллюстрировано рис. 1. Так как точка измерения представляется в виде круглой области, достаточно велика вероятность попадания этой области на границу ячеек, в таком случае сложно сделать вывод о принадлежности точки к той или иной ячейке, поэтому делается допущение, что точка принадлежит сразу к нескольким ячейкам, и ее значение уровня сигнала в этой точке соответственно участвует в вычислении средних уровней всех ячеек, к которым она принадлежит.

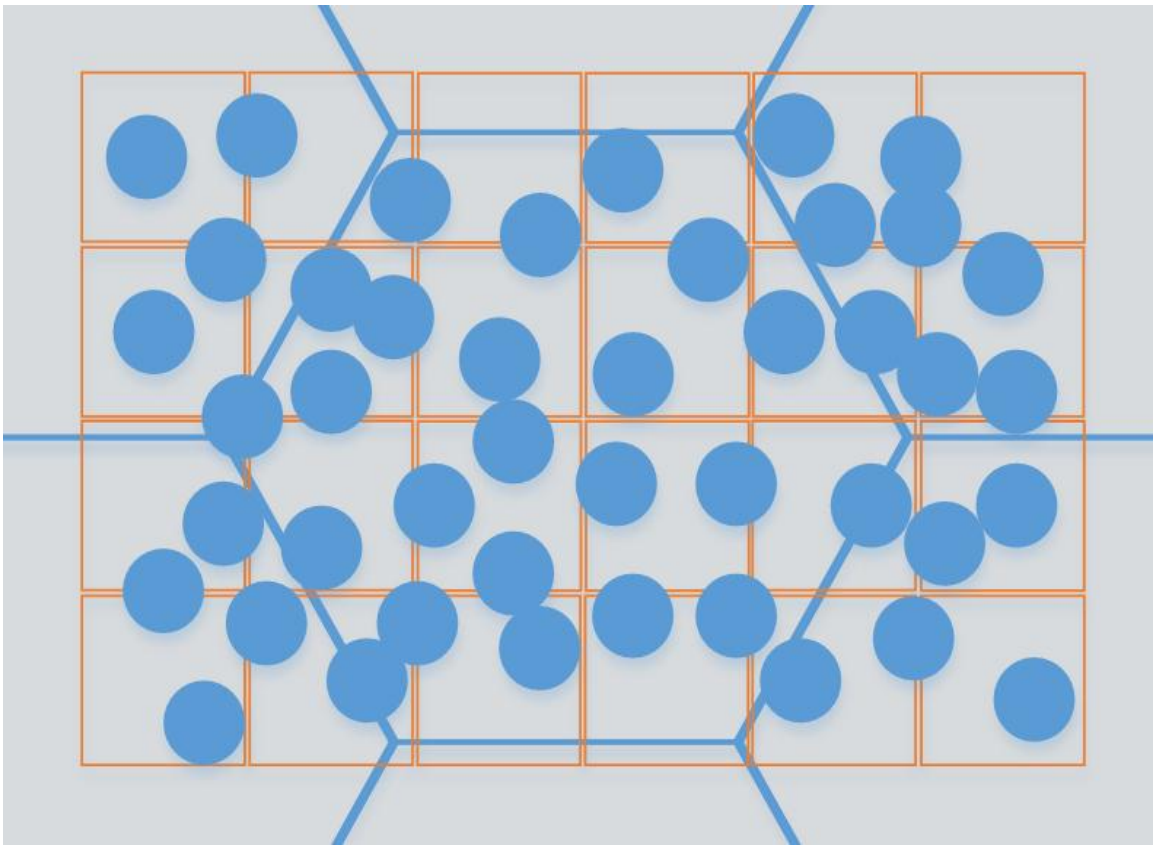


Рис. 1. Точки измерения технических параметров, распределенные
по дискретным ячейкам карты покрытия

Таким образом, реальная карта покрытия представляется как совокупность дискретных ячеек, с рассчитанными средними значениями уровня сигнала, а актуализация карты покрытия осуществляется за счет периодического обновления усредненных значений уровня сигнала в ячейках, полученных на основе новых данных измерений, присылаемых абонентами. В настоящий момент в рамках проекта программы «УМНИК» ведется разработка программного комплекса, использующего описанную методологию построения карт покрытия, и данная работа также написана в рамках этого проекта.

Библиографический список

1. Ханин, И. В. Информационно-измерительная система для мониторинга сетей мобильной связи : дис. ... канд. техн. наук (05.11.16. Пензенский государственный университет) / Ханин И. В. – Пенза, 2012.

2. Мартяшин, Г. В. Проблемы диагностики и мониторинга технических параметров сетей сотовой связи / Г. В. Мартяшин, И. С. Пышкина, А. В. Калачев // Взаимодействие науки и общества: проблемы и перспективы : сб. ст. Междунар. науч.-практ. конф. : в 4 ч. (г. Казань, 8 июня 2017 г.). – Казань, 2017. – Ч. 3. – С. 74–76.

3. Principles of cellular frequency reuse // JPL's Wireless Communication Reference Website [сайт]. – URL: http://www.wirelesscommunication.nl/reference/chaptr04/cellplan/reuse.htm (дата обращения: 16.10.2016).

4. Анучин, О. Н. Интегрированные системы ориентации для морских подвижных объектов / О. Н. Анучин, Г. И. Емельянцев ; под ред. В. Г. Пешехонова. – 2-е изд. – СПб. : ГНЦ РФ-ЦНИИ «Электроприбор», 2003. – С. 160–161.

5. Вентцель, Е. С. Теория вероятностей : учеб. для вузов / Е. С. Вентцель. – 6-е изд., стер. – М. : Высш. шк., 1999. – 576 с.

6. Левин, Б. Р. Теоретические основы статистической радиотехники / Б. Р. Левин. – 3-е изд., перераб. и доп. – М. : Радио и связь, 1989. – 656 с.

Г. В. Мартяшин, И. С. Пышкина

Пензенский государственный университет, Пенза, Россия

АНАЛИТИЧЕСКИЙ ПРОГРАММНЫЙ ИНСТРУМЕНТ ПЛАНИРОВАНИЯ ЭКСПЕРИМЕНТА

Обсуждается вопрос разработки составов строительных смесей путем планирования эксперимента с помощью специальных программных средств. Рассматривается разрабатываемое аналитическое программное средство, которое является более простой и удобной альтернативой существующим решениям.

В настоящий момент активно развивается сфера разработки отечественных строительных материалов, о чем можно судить и по достаточно большому количеству научных работ на подобные темы, включающих публикации, патенты и диссертации.

Во многом это определяется повышением спроса на строительные материалы по причине увеличения объемов строительства в России [1]. Наибольшее распространение на отечественном рынке имеют импортные строительные материалы или отечественными строительными материалами, которые основаны на импортных составляющих. В то же время, использование отечественных аналогичных строительных материалов, особенно в контексте импортозамещения, имеет ряд существенных преимуществ по сравнению с импортными. Во-первых, использование отечественных материалов делает рынок менее импортозависимым, укрепляя тем самым российскую экономику. Во-вторых, отечественные материалы обладают более низкой себестоимостью, за счет колоссального сокращения издержек на транспортировку при использовании местных сырьевых ресурсов.

Строительные материалы как правило имеют рецептуры, включающие большое количество компонентов, причем конченые свойства материала определяется соотноше-нием массовых долей компонентов, их химическим составом, режимом синтеза. Например, сухие строительные смеси (ССС), занимающие немалую часть рынка, имеют рецептуры, в которые входят вяжущие, заполнитель, минеральные, пластифицирующие, гидрофобизирующие и полимерные добавки [2].

Разработка строительных материалов, таким образом, подразумевает решение задачи поиска оптимального соотношения каких-либо количественных параметров для получения требуемых свойств конечного продукта. Такими параметрами могут быть, например, массовые доли компонентов смеси и температура твердения. Это означает, что подобного рода исследование опирается на эксперимент, как на основной метод. Так как проведение экспериментов требует дополнительных материальных и временных ресурсов (продолжительность эксперимента может достигать нескольких месяцев), количество необходимых экспериментов всегда требуется сводить к минимальному. Для этого прибегают к такому методу, как планирование эксперимента. Метод позволяет построить математическую модель зависимости между оптимизируемыми свойствами смеси и количественно выраженными факторами, а затем, после проведения минимального необходимого количества экспериментов, опровергнуть или же подтвердить и уточнить модель. Затем определяется оптимальное решение (оптимальные значения количественных факторов) на основе поиска экстремумов модели [3].

В настоящие время планирование эксперимента, как и любая обработка результатов экспериментов без использования компьютеров и пакетов прикладных программ практически не применяется в силу своей рутинности. Наиболее универсальным из подобных средств являются табличные процессоры с различными математическими надстройками и математические программные пакеты, например, MATLAB. Однако, в силу их универсальности подобных средства не ориентированы на конкретные сценарии, и их использование предполагает знание определенных особенностей и наличия некоторых навыков програм-

мирования, что делает их применение весьма проблематичным для тех исследователей, область научных интересов которых далека от информационных технологий. Этим продиктована необходимость разработки более простых в использовании узкоспециализированных средств [4].

В настоящий момент в этих целях на завершающих этапах находится разработка аналитического программного средства для планирования эксперимента и обработки результатов экспериментов, ориентированное для применения в сфере разработки строительных материалов. В качестве программной платформы для разработки выбрана .NET Framework.

Разрабатываемое программное средство может быть проиллюстрировано UML-диаграммой вариантов использования, которая представлена на рис. 1.



Рис. 1. Диаграмма вариантов использования

Из данной диаграммы видно, что присутствуют все прецеденты, необходимые для выполнения планирования эксперимента. В качестве начальных вариантов использования выступает «планирование эксперимента», «сохранение результатов планирования» и «загрузка сохраненных результатов планирования». Вариант использования «планирование эксперимента», соответствующий началу нового планирования эксперимента, связан отношениями включения и расширения с другими вариантами использования, относящимися к более глубокому уровню детализации модели. Так, «указание факторов и типа модели», «формирование матрицы планирования эксперимента», «ввод полученных экспериментальных значений» являются прецедентами, включаемыми в состав начала нового планирования эксперимента. Отношение включения подразумевает обязательное наличие данных вариантов использования в рамках начала нового планирования эксперимента, как неотъемлемых его частей. Варианты использования «расчет математических моделей» и «формирование графиков» связаны с началом нового планирования эксперимента отношением расширения, так как выступают как действия, выполняющиеся на последующих этапах планирования эксперимента, то есть в начале планирования являются необязательными.

Сохранение результатов может быть выполнено на любом этапе планирования и не связано с какими-либо отношениями с другими вариантам использования.

Загрузка сохраненных результатов планирования эксперимента расширяется такими вариантами использования, как «ввод полученных экспериментальных значений», «расчет математической модели» и «формирование графиков». Отношение расширения объясняется тем, что сохраненные результаты планирования могут соответствовать любому этапу

планирования, т.е. это может быть начальный этап, на котором только определены факторы и тип модели, и сгенерирована матрица планирования, так и завершающий этап, на котором уже построена математическая модель и оценена ее адекватность. Так сохраненные результаты планирования могут соответствовать разным этап, все дальнейшие варианты использования рассматриваются как необязательные.

Итак, типичный сценарий работы с разрабатываемым программным инструментом имеет следующий вид:

- начинается новое планирование эксперимента, после получения матрицы планирования результаты сохраняются;
- по мере выполнения экспериментов в соответствии с матрицей планирования периодически производится загрузка результатов, ввод полученных в результате экспериментов значений и сохранение результатов;
- после проведения всех необходимых экспериментов строится математическая модель и оценивается ее адекватность, а также производится формирование графиков, иллюстрирующих полученную математическую модель.

Таким образом, была рассмотрена проблема планирования эксперимента и обработки экспериментальных данных при разработке строительных материалов. Были предложено решение в виде аналитического программного средства, решающего поставленную проблему и удовлетворяющего поставленным требованиям. В дальнейшем планируется завершение разработки описанного в данной статье программного инструмента.

Библиографический список

1. Савченко, Е. Ю. Рынок строительных материалов: тенденции развития / Е. Ю. Савченко // Экономика, управление и право: инновационное решение проблем : сб. ст. победителей VI Междунар. науч.-практ. конф. / под общ. ред. Г. Ю. Гуляева. – Пенза : МЦНС «Наука и Просвещение», 2017. – С. 149–153.

2. Оптимизация режима синтеза добавки на основе гидросиликатов для известковых сухих строительных смесей / В. И. Логанина, И. С. Пышкина, С. А. Толушов, С. А. Болдырев // Региональная архитектура и строительство. – 2015. – № 4-1 (25). – С. 48–51.

3. Логанина, В. И. Разработка технологии синтеза добавки на основе гидросиликатов кальция для сухих строительных смесей / В. И. Логанина, И. С. Пышкина // Известия высших учебных заведений. Строительство. – 2014. – № 7 (667). – С. 46–49.

4. Мартяшин, Г. В. Программный комплекс планирования эксперимента и анализа экспериментальных данных / Г. В. Мартяшин, И. С. Пышкина // Новые информационные технологии и системы (НИТиС – 2015) : сб. науч. ст. XII Междунар. науч.-техн. конф. – Пенза : Изд-во ПГУ, 2015. – С. 63–65.

С. В. Тумасов, Е. И. Гурин

Пензенский государственный университет, Пенза, Россия

АНАЛИЗАТОР КАЧЕСТВА СВЯЗИ НА БАЗЕ ШУМОПОДОБНОГО СИГНАЛА

Представлено программно-аппаратное решение для оценки качества связи на базе шумоподобного сигнала.

Беспроводные технологии в настоящее время применяются во многих прикладных областях, таких как системы обеспечения безопасности и обороны, управления энергосбережением, охранные системы, мониторинг состояния человека. Особо важной становится проблема организации беспроводной сети в сложных условиях, например, в лесу. Одним из вариантов решения этой проблемы является применение технологии шумоподобного сиг-

нала. Она позволяет увеличить дальность связи по сравнению с узкополосными сигналами. При этом важно выполнить оценку качества связи [1].

В системах LPWAN (энергоэффективная сеть дальнего радиуса действия) широко применяется стандарт LoRaWAN. Технология LoRa реализует физический уровень системы, представляет собой схему модуляции с расширением спектра, в которой данные кодируются широкополосными ЛЧМ-импульсами с частотой, увеличивающейся или уменьшающейся на некотором временном интервале (метод CSS). Метод CSS обеспечивает больший бюджет канала, защищен от эффектов многолучевого распространения, эффекта Допплера. При этом в LoRa применяются коды Грея, расширение спектра путем выполнения операции «ИСКЛЮЧАЮЩЕЕ ИЛИ» над данными и псевдослучайной последовательностью, используются коды Хэмминга прямой коррекции ошибки [2].

В настоящей работе рассматривается анализатор качества связи для LoRa. Он представляет собой макеты передатчика и приемника LoRa. Макет приемника имеет в своем составе одноплатный компьютер Raspberry Pi, приемник/передатчик на базе SX1278, GPS-приемник, LCD-экран. Макет передатчика включает в себя одноплатный компьютер Raspberry Pi, приемник/передатчик на базе SX1278. Было разработано программное обеспечение для передатчика и приемника, реализующее функционал анализатора качества связи. Структурная схема макетов приемника и передатчика представлена ниже.

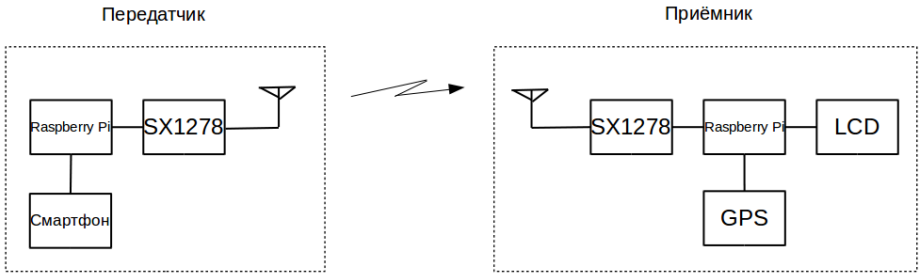


Рис. 1. Структурная схема макетов приемника и передатчика

В качестве используемого диапазона для работы рассматривались LPD433 (433 МГц), 868 (UHF)(868 МГц), 2400(S) (2400 МГц). Был выбран диапазон LPD433, потому что он обеспечивает наибольшую возможную дальность работы при одной и той же мощности [3].

В ходе эксперимента передатчик создавал точку доступа Wi-Fi. К этой точке доступа подключался смартфон. Далее через терминал устанавливалась требуемая длина псевдослучайной последовательности, скорость передачи [4]. Для экспериментов использовались две комбинации параметров:

1) для передачи SMS устанавливалась комбинация: 7,8 КГц – скорость передачи, 4096 чипов – длина псевдослучайной последовательности;

2) для передачи голоса устанавливалась комбинация: 500 КГц – скорость передачи, 128 чипов – длина псевдослучайной последовательности.

После чего Raspberry Pi в макете передатчика формировал пакеты, а передатчик/приемник на базе SX1278 передавал их. В макет приемника вводились такие же установки с помощью LCD-экрана с тачскрином. Если параметры введены верно, приемник обнаруживал передаваемый от передатчика сигнал. Для анализа качества связи было решено подключить к Raspberry Pi в макете приемника GPS-приемник для записи треков. GPS-приемник отправляет координаты местоположения макета приемника каждую секунду Raspberry Pi. При успешном получении пакета от передатчика в макете приемника Raspberry Pi записывает координаты в лог-файл в определенном формате. В ходе эксперимента этот лог-файл заполняется значениями. Для каждой комбинации параметров создается собственный лог-файл. После эксперимента полученный лог-файл транслируется в kml-файл, который позволяет посмотреть трек в программе Google Earth. Треки анализируются и выбирается максимальная возможная дальность связи.

Фотография макетов передатчика и приемника представлена ниже.

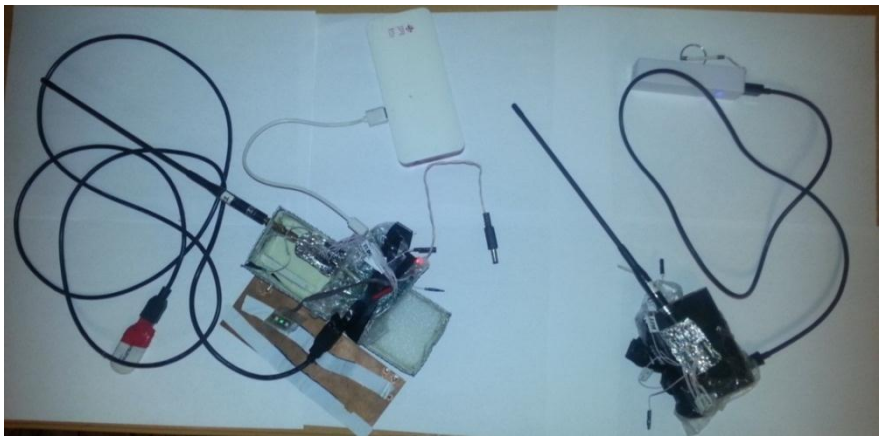


Рис. 2. Макеты приемника и передатчика

Эксперимент проводился в лесистой местности. На передатчике запускалась передача и он находился в статическом состоянии, а приемник перемещался и записывал логфайл. При комбинации параметров для передачи SMS максимальная возможная дальность связи составила 758 м, а для голоса она составила 342 м.

Предложенные авторами программно-аппаратные решения позволят проанализировать качество связи на базе шумоподобного сигнала, измеряя максимально возможную дальность связи.

Библиографический список

1. Тумасов, С. В. Макет сенсорной сети с использованием технологии SDR / С. В. Тумасов, Е. К. Таранцев // Информационные технологии в науке и образовании : сб. ст. II Междунар. науч.-техн. конф. – Пенза : Изд-во ПГУ, 2015. – С. 304–308.

2. Matt Knight Reversing LoRa : Exploring next-generation wireless. – Bastille Networks, 2016. – 128 с.

3. Таранцев, Е. К. Сеть радиомаяков с расширенным спектром / Е. К. Таранцев, С. В. Тумасов // XXI век: итоги прошлого и проблемы настоящего плюс. – 2016. – № 3 (31). – С. 161–166.

4. Miguel, Elias M. Campista, Marcelo G. Rubinstein Advanced Routing Protocols for Wireless Networks / Miguel Elias M. Campista, Marcelo G. – UK : Wiley, 2014.

К. И. Никишин

Пензенский государственный университет, Пенза, Россия

АППАРАТНАЯ КОНФИГУРАЦИЯ ПЛИС ТИПА FPGA С ИСПОЛЬЗОВАНИЕМ ИНТЕРФЕЙСА ETHERNET ДЛЯ ОБМЕНА ИНФОРМАЦИЕЙ

Рассматривается аппаратная конфигурация ПЛИС типа FPGA с использованием интерфейса Ethernet для обмена информацией на основе системы на кристалле (System on a Chip) в среде Xilinx. Конфигурация ПЛИС использует процессорное ядро MicroBlaze, через которое подключаются все необходимые периферийные устройства.

В настоящее время технология Ethernet является самой распространенной технологией в мире. При разработке новых типов конфигурации Ethernet, новых алгоритмов дис-

петчеризации [1] в коммутаторе большое внимание уделяется к параметрам быстродействия коммутатора, минимизации аппаратных затрат, легкости встраиваемости системы.

Для корректной отладки интерфейса Ethernet используют различные программные пакеты, позволяющие моделировать различные аспекты поведения сложных телекоммуникационных систем [2]. К такому программному пакету относится CPN Tools [3]. Для проверки реализованной модели в CPN Tools на аппаратуре можно использовать программируемые логические интегральные схемы (ПЛИС) типа FPGA в среде Xilinx [4].

Конфигурация ПЛИС использует процессорное ядро *MicroBlaze*, через которое подключается все необходимые периферийные устройства. В данном случае подключается интерфейс Ethernet для обмена информацией на основе системы на кристалле (System on a Chip) [5]. В статье используется отладочная плата Spartan 3E Starter Kit (рис. 1) для проверки аппаратных и программных особенностей [6].

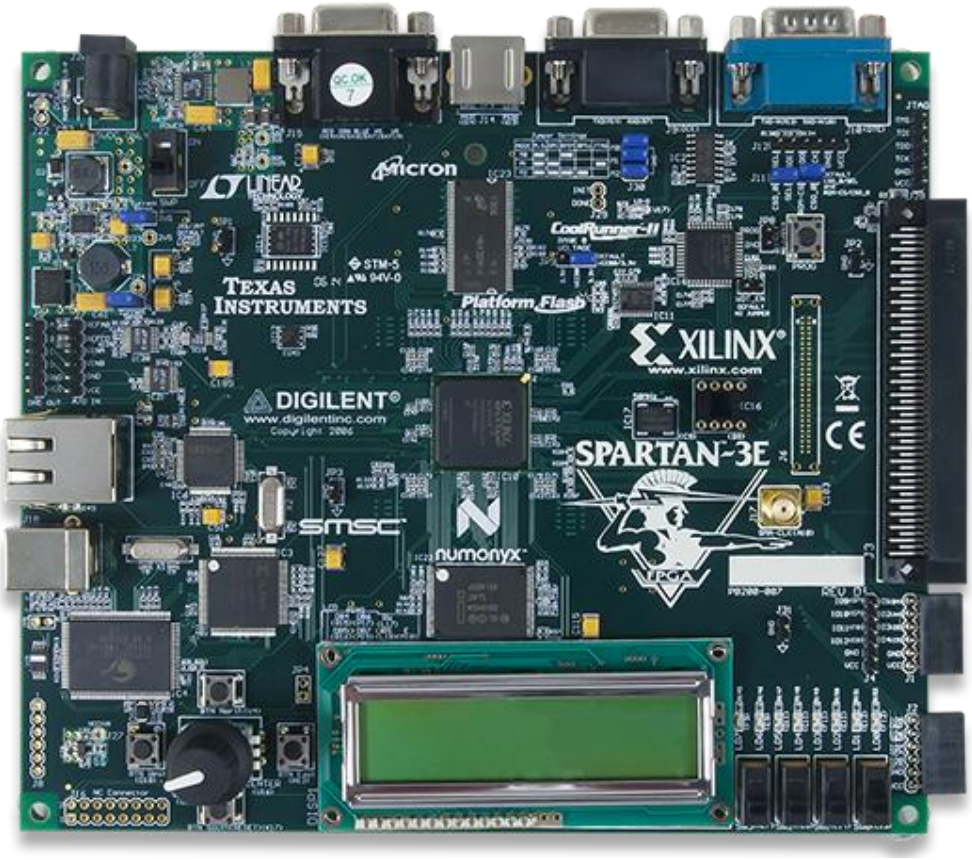


Рис. 1. Отладочная плата Spartan 3E Starter Kit

При создании проекта в XPS (Xilinx Platform Studio) задавался следующий тип микросхемы ПЛИС семейства Spartan 3E: *XC3S500E-4FG320*, полярность сброса (Reset polarity – Active HIGH), выбиралось процессорное ядро *MicroBlaze* и настраивались его необходимые характеристики (в частности частота процессорного ядра). В проекте использовались следующие внешние периферийные устройства: переключатели DIP_Switches на 4 разряда, светодиоды LEDs на 8 разрядов, интерфейс RS232 для вывода информации через COM-порт и интерфейс Ethernet для проверки обмена информации.

После создания проекта должны создаться основные аппаратные спецификации для обеспечения работы отладочной платы. Спецификация аппаратной платформы встраиваемой микропроцессорной системы MHS (*Microprocessor Hardware Specification*) представляет собой файл с расширением MHS, в котором в текстовом формате описывается конфигурация и параметры аппаратных средств проектируемой системы [7].

Дополнительная информация, используемая программами синтеза, размещения и трассировки проекта аппаратной части микропроцессорной системы, задается в форме файла временных и топологических ограничений *User Constraints File* (UCF) [8]. После окончания работы мастера BSB на экране появляется основное окно XPS (Xilinx Platform Studio), верхняя часть которого показана на рис. 2.

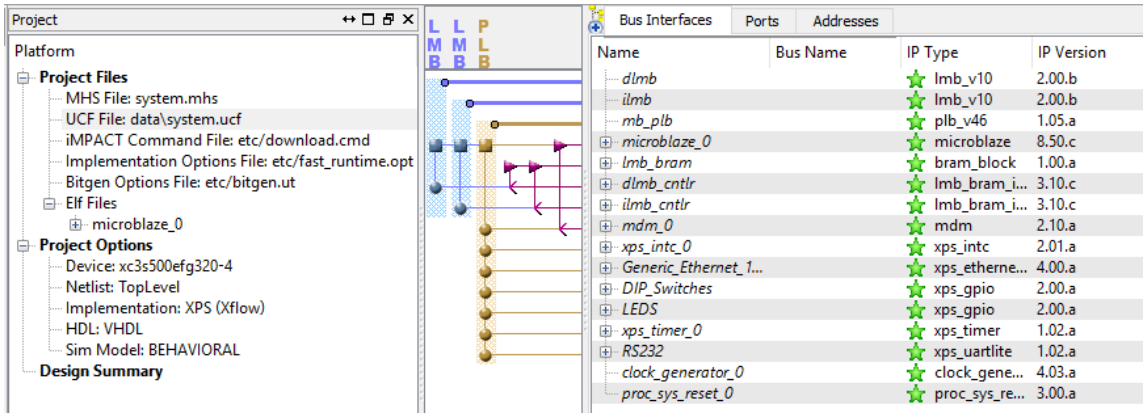


Рис. 2. Основное окно XPS (Xilinx Platform Studio)
после установки необходимых параметров проекта

Компонент интерфейса Ethernet, включаемый в состав спецификации аппаратной платформы, выглядит следующим образом:

BEGIN xps_ethernetlite
PARAMETER INSTANCE = Generic_Ethernet_10_100
PARAMETER HW_VER = 4.00.a
PARAMETER C_BASEADDR = 0x81000000
PARAMETER C_HIGHADDR = 0x8100ffff
BUS_INTERFACE SPLB = mb_plb
PORT PHY_tx_clk = fpga_0_Generic_Ethernet_10_100_PHY_tx_clk_pin
PORT PHY_rx_clk = fpga_0_Generic_Ethernet_10_100_PHY_rx_clk_pin
PORT PHY_crs = fpga_0_Generic_Ethernet_10_100_PHY_crs_pin
PORT PHY_dv = fpga_0_Generic_Ethernet_10_100_PHY_dv_pin
PORT PHY_rx_data = fpga_0_Generic_Ethernet_10_100_PHY_rx_data_pin
PORT PHY_col = fpga_0_Generic_Ethernet_10_100_PHY_col_pin
PORT PHY_rx_er = fpga_0_Generic_Ethernet_10_100_PHY_rx_er_pin
PORT PHY_rst_n = fpga_0_Generic_Ethernet_10_100_PHY_rst_n_pin
PORT PHY_tx_en = fpga_0_Generic_Ethernet_10_100_PHY_tx_en_pin
PORT PHY_tx_data = fpga_0_Generic_Ethernet_10_100_PHY_tx_data_pin
PORT PHY_MDC = fpga_0_Generic_Ethernet_10_100_PHY_MDC_pin
PORT PHY_MDIO = fpga_0_Generic_Ethernet_10_100_PHY_MDIO_pin
END

Значение базового адреса компонента определяется с помощью параметра *C_BASEADDR*. Верхняя граница адресного пространства памяти, используемого компонентом, указывается в виде значения параметра *C_HIGHADDR*. Диапазон адресов можно настроить и на вкладе проекта адресов «*Addresses*». Пользователь может сам установить необходимый диапазон вручную или нажать на кнопку автоматической генерации адресов «*Generate Addresses*».

Внешние порты компонента: *PHY_tx_clk* – порт синхросигнала передатчика; *PHY_rx_clk* – порт синхросигнала приемника; *PHY_crs* – порт чувствительности; *PHY_dv* – порт корректности принимаемой информации; *PHY_rx_data* – порт принимаемой информации; *PHY_col* – порт определения коллизий; *PHY_rx_er* – порт ошибки приема; *PHY_rst_n* – порт сброса; *PHY_tx_en* – порт разрешения передачи; *PHY_tx_data* – порт передаваемой

информации; *PHY_MDC* – синхросигнал управления; *PHY_MDIO* – управление входной/выходной информацией.

Важной частью проекта является файл ограничений, который позволяет установить соответствие между внешними цепями проектируемой системы и номерами выводов ПЛИС. Для создания файла временных и топологических ограничений и внесения в него информации можно использовать встроенный текстовый редактор управляющей оболочки Xilinx Platform Studio.

Ограничения для интерфейса Ethernet выглядят следующим образом:

Net fpga_0_Generic_Ethernet_10_100_PHY_tx_clk_pin LOC=T7;
Net fpga_0_Generic_Ethernet_10_100_PHY_rx_clk_pin LOC=V3;
Net fpga_0_Generic_Ethernet_10_100_PHY_crs_pin LOC=U13;
Net fpga_0_Generic_Ethernet_10_100_PHY_dv_pin LOC=V2;
Net fpga_0_Generic_Ethernet_10_100_PHY_rx_data_pin<0> LOC=V8;
Net fpga_0_Generic_Ethernet_10_100_PHY_rx_data_pin<1> LOC=T11;
Net fpga_0_Generic_Ethernet_10_100_PHY_rx_data_pin<2> LOC=U11;
Net fpga_0_Generic_Ethernet_10_100_PHY_rx_data_pin<3> LOC=V14;
Net fpga_0_Generic_Ethernet_10_100_PHY_col_pin LOC=U6;
Net fpga_0_Generic_Ethernet_10_100_PHY_rx_er_pin LOC=U14;
Net fpga_0_Generic_Ethernet_10_100_PHY_tx_en_pin LOC=P15;
Net fpga_0_Generic_Ethernet_10_100_PHY_tx_data_pin<0> LOC=R11;
Net fpga_0_Generic_Ethernet_10_100_PHY_tx_data_pin<1> LOC=T15;
Net fpga_0_Generic_Ethernet_10_100_PHY_tx_data_pin<2> LOC=R5;
Net fpga_0_Generic_Ethernet_10_100_PHY_tx_data_pin<3> LOC=T5;

Таким образом, видно соответствие наименований портов в спецификации аппаратной платформы и в файле ограничений. В файле ограничений указывается еще и разрядность, порты принимаемой/передаваемой информации являются четырехразрядными.

В статье рассмотрена аппаратная конфигурация ПЛИС типа FPGA с использованием интерфейса Ethernet для обмена информацией на основе системы на кристалле (System on a Chip) в среде Xilinx. Все основные компоненты (периферийные устройства) ПЛИС подключались через процессорное ядро *MicroBlaze*. Рассмотрены спецификация аппаратной платформы (MHS) и файл временных и топологических ограничений (UCF) для интерфейса Ethernet на плате Spartan 3E Starter Kit.

Библиографический список

1. Никишин К. И. Механизм управления трафиком реального времени в коммутаторе Ethernet / К. И. Никишин // Вестник компьютерных и информационных технологий. – 2015. – Вып. 10. – С. 32–37.

2. Scheduling queues in the Ethernet switch, considering the waiting time of frames / Evgeniy Kizilov, Nikolas Konnov, Kirill Nikishin, Dmitry Pashchenko, Dmitriy Trokoz // MATEC Web of Conferences . – 2016. – Vol. 44. – P. 01011-p.1-01011-p.

3. Описание программы CPN Tools. – URL: http://cpntools.org/

4. Описание работы с программными средами фирмы Xilinx. – URL: http://xilinx.com/

5. Никишин, К. И. Разработка выходного порта коммутатора Ethernet на базе ПЛИС / К. И. Никишин // Информационные технологии. Проблемы и решения : сб. материалов III Междунар. науч.-практ. конф. – Уфа : Восточная печать, 2015. – Т. 2. – С. 201–204.

6. Руководство к плате Spartan-3E Start Kit Board User Guide UG230 2006.

7. Зотов, В. Формирование спецификации аппаратной платформы микропроцессорной системы на основе ядра MicroBlaze, реализуемой в ПЛИС семейств FPGA фирмы Xilinx / В. Зотов // Компоненты и технологии. – 2004. – № 7-8.

8. Зотов, В. Реализация аппаратной платформы микропроцессорной системы, проектируемой на основе ядра MicroBlaze, в ПЛИС семейств FPGA фирмы Xilinx / В. Зотов // Компоненты и технологии. – 2004. – № 9.

К. И. Никишин

Пензенский государственный университет, Пенза, Россия

ОБМЕН ИНФОРМАЦИЕЙ С ИСПОЛЬЗОВАНИЕМ ИНТЕРФЕЙСА ETHERNET НА ОСНОВЕ ПЛИС ТИПА FPGA

Рассматривается обмен информацией с использованием интерфейса Ethernet на основе программируемых логических интегральных схем ПЛИС типа FPGA. Интерфейс Ethernet подключается через процессорное ядро MicroBlaze.

В настоящее время технология Ethernet является самой распространенной технологией в мире. Большое внимание при разработке улучшений Ethernet уделяется быстродействию, повышению качества обслуживания сети. Моделирование всей системы Ethernet позволяет избежать многих ошибок и выявить закономерности в обработки системы [1, 2]. Используются различные алгоритмы диспетчеризации очередей в коммутаторе Ethernet и выявляют особенности обработки трафика [2].

Конфигурация ПЛИС использует процессорное ядро *MicroBlaze*, через которое подключается все необходимые периферийные устройства. В данном случае подключается интерфейс Ethernet для обмена информацией на основе системы на кристалле (System on a Chip) [3]. В статье используется отладочная плата Spartan 3E Starter Kit для проверки аппаратных и программных особенностей [4].

Обмен данными по каналу Ethernet осуществляется между ПК и отладочной платой Spartan 3E Starter Kit. Для работы платы Spartan 3E Starter Kit на уровне интерфейса Ethernet в системе Xilinx SDK был создан проект для данной схемы, в котором помимо стандартных периферийных устройств был подключен контроллер Ethernet [5]. Формат кадра Ethernet представлен на рис. 1.

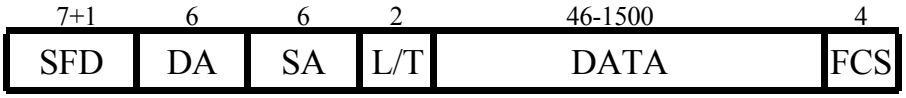


Рис. 1. Кадр Ethernet

Рассмотрим реализацию обмена данных с использованием интерфейса Ethernet в среде SDK [6–7]. Вначале работы с контроллером Ethernet устанавливается указатель на драйвер контроллера.

*XEmacLite *EmacLiteInstancePtr = &EmacLiteInstance;*

После установления указателя на котроллер необходимо проинициализировать Ethernet. За инициализацию отвечает функция *XEmacLite_Initialize*. Функция возвращает статус успешности или неудачи инициализации контроллера.

*int XEmacLite_Initialize(XEmacLite *InstancePtr, u16 DeviceId)*, где *InstancePtr* – указатель на контроллер, *DeviceId* – идентификатор устройства.

После инициализации контроллера необходимо установить MAC-адрес отладочной платы через функцию *XEmacLite_SetMacAddress*, входными параметрами функции являются указатель на контроллер и необходимое значение MAC-адреса.

После выполняется функция очистки буферов приема. Через функцию *XEmacLite_TxBufferAvailable* определяются, заполнены ли буферы передачи и доступны ли

они для дальнейшей обработки. Если возвращается истинное значение, значит, отладочная плата успешно приняла информацию от ПК.

Функция получения кадра (поступившей информации) выглядит следующим образом:

*u16 XEmacLite_Recv(XEmacLite *InstancePtr, u8 *FramePtr)*, где первым параметром выступает указатель на контроллер, второй параметр – указатель на буфер, который будет хранить кадр. Функция возвращает 0 в случае, если нет информации в буфере.

Для двустороннего взаимодействия программа работы интерфейса Ethernet была сделана таким образом, что при получении кадра по адресу 169.254.255.255 на ПК отправлялся ответный пакет, подтверждающий прием команды PING. Для фиксации входящего трафика использовался сниффер Microsoft Network Monitor.

```
if((*(Xuint8 *)(RxFrame + 30) == 169) &&
(*(Xuint8 *)(RxFrame + 31) == 254) &&
(*(Xuint8 *)(RxFrame + 32) == 255) &&
(*(Xuint8 *)(RxFrame + 33) == 255)) {}
```

После успешного сравнения полученного ответа по заданному адресу выполняется алгоритм отправки кадра. Сначала устанавливается адрес назначения (адрес ПК), затем записывается адрес источника (адрес отладочной платы) в кадр передачи. Устанавливается поле длины кадра и затем заполняется поле данных кадра передачи. После установки всех необходимых полей кадра передачи вызывается функция *XEmacLite_Send*.

*int XEmacLite_Send(XEmacLite *InstancePtr, u8 *FramePtr, unsigned ByteCount)*, где *InstancePtr* - указатель на контроллер, *FramePtr* – указатель на буфер передачи, *ByteCount* – размер кадра в байтах.

После отправки команды PING по сети передавались пакеты. Во время обработки данных, осуществлялась отправка ответного сообщения. Об отправке пакета данных свидетельствует мигание диода выходного интерфейса Ethernet.

Принятые пакеты фиксируются сниффером Microsoft Network Monitor (рис. 2). Как видно из рис. 2 в сеть отправляется сообщение с адресом 169.254.255.255 в ответ, на которое приходит сообщение, отправленное отладочной платой.

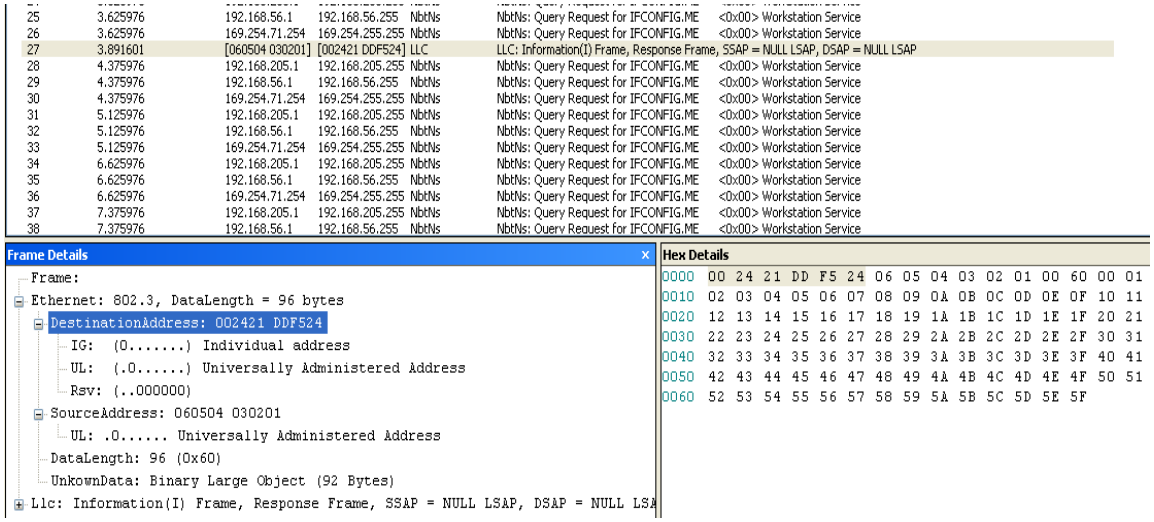


Рис. 2. Входящий трафик

В статье рассмотрен программный обмен информацией на основе системы на кристалле (System on a Chip) с использованием интерфейса Ethernet в среде Xilinx. Интерфейс Ethernet подключался через процессорное ядро MicroBlaze. Рассмотрены основные функции обработки интерфейса Ethernet, функции работы с кадрами принятой и отправленной информации на плате Spartan 3E Starter Kit.

Библиографический список

1. Scheduling queues in the Ethernet switch, considering the waiting time of frames / Kizilov Evgeniy, Konnov Nikolas, Nikishin Kirill, Pashchenko Dmitry, Trokoz Dmitriy // MATEC Web of Conferences. – 2016. – Vol. 44. – P. 01011-p.1-01011-p.
2. Никишин, К. И. Механизм управления трафиком реального времени в коммутаторе Ethernet / К. И. Никишин // Вестник компьютерных и информационных технологий. – 2015. – Вып. 10. – С. 32–37.
3. Никишин, К. И. Разработка выходного порта коммутатора Ethernet на базе ПЛИС / К. И. Никишин // Информационные технологии. Проблемы и решения : сб. материалов III Междунар. науч.-практ. конф. – Уфа : Восточная печать, 2015. – Т. 2. – С 201–204.
4. Руководство к плате Spartan-3E Start Kit Board User Guide UG230 2006.
5. Описание работы с программными средами фирмы Xilinx. – URL: http://xilinx.com/
6. Зотов, В. Разработка программного обеспечения микропроцессорной системы на основе ядра MicroBlaze, реализуемой в ПЛИС семейств FPGA фирмы Xilinx / В. Зотов // Компоненты и технологии. – 2005. – № 2.
7. Зотов, В. Разработка встраиваемых микропроцессорных систем на основе ядра MicroBlaze, реализуемых в ПЛИС семейств FPGA фирмы Xilinx, с помощью «мастера» Base System Builder Wizard / В. Зотов // Компоненты и технологии. – 2005. – № 3-4.

А. О. Семенов

Пензенский государственный университет, Пенза, Россия

УСТРОЙСТВО, РЕАЛИЗУЮЩЕЕ АЛГОРИТМ ДИСПЕТЧЕРИЗАЦИИ НА БАЗЕ АДАПТИВНЫХ RAND-ПРИНЦИПОВ

Приводится описание особенностей построения модели алгоритма стохастической диспетчеризации очередей в составе сетевых коммутаторов с использованием сетей Петри.

В настоящей работе предлагается описание структуры сетей Петри, реализующих различные модификации стохастического алгоритма диспетчеризации, основанного на стохастическом подходе к выбору очередей на обслуживание с адаптацией к длинам очередей и текущему трафику в отдельных классах QoS.

При разработке устройства, обеспечивающего работу заданного алгоритма диспетчеризации, в составе сетевого коммутатора следует разработать, прежде всего, архитектуру построения коммутатора, в том числе исходя из понятий модульности, расширяемости и др. [1–4] С этой целью в качестве инструмента создания модели сетевого коммутатора взято средство создания иерархических сетей Петри.

В работе [2] рассматривалась разновидность алгоритма, на базе принципов хаотического выбора очередей, при котором каждая очередь имеет пакеты различной длины. Также был предложен вариант имитационной модели устройства, моделирующего стохастический алгоритм. В рамках данной работ

В состав устройства стохастической диспетчеризации очередей сетевого коммутатора с поддержкой качества обслуживания, содержит: генератор псевдослучайных чисел; узел хранения постоянных величин (ПЗУ); узел подсчета вероятностей выбора очередей на обслуживание Узел сравнения величин; узел подсчета номера обслуживаемой очереди. На рис. 1 показана подсеть диспетчера, принимающая на свои входы длины очередей, а на выходы поступает единственная метка обслуживания очереди.

В состав модели подсети, изображенной на рис. 1 входят узел сбора (надзора) входной информации (1, 3), генератор псевдослучайных чисел (2, 11); узел хранения постоянных величин (ПЗУ) (4); узел подсчета вероятностей выбора очередей на обслуживание (5, 6), узел сравнения величин (7–13); узел подсчета номера обслуживаемой очереди (14, 15).

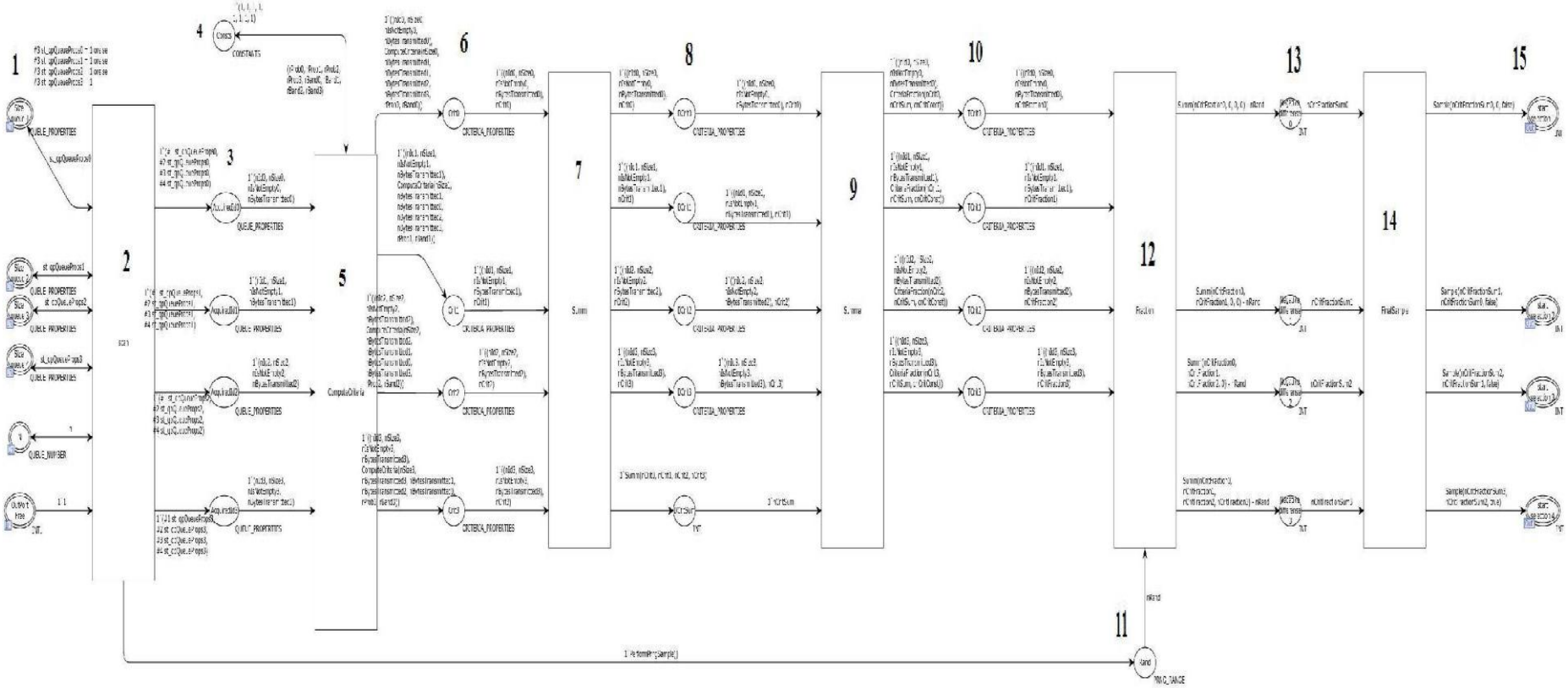


Рис. 1. Подсеть Петри, реализующая стохастической выбор очереди

Устройство стохастической диспетчеризации в составе коммутатора или маршрутизатора предназначено для непосредственного управления и надзора за процессом распределения канального времени между разными очередями. Узел надзора осуществляет контроль над выполнением процесса обработки и продвижения кадров кадровым обработчиком. Если обслуживание одного из классов трафика не укладывается по времени в нормативы, определяемые как количество квантов времени, выделяемых узлом надзора на обслуживание и передачу в канал из передаточной очереди кадра заданного размера, то такое обслуживание заканчивается узлом надзора принудительно. Таким образом, реализуется одно из базовых правил массового обслуживания.

Процесс расчета номера обслуживаемой очереди является циклическим, запускается блоком выборки кадра (1, 2). Передача аргументов для подсчета вероятностей осуществляется из передаточных очередей заголовков и указателей кадрового обработчика к подсети стохастической диспетчеризации. Узел надзора проверяет аргументы на корректность и разрешает или запрещает расчет. Подсчет вероятностей для каждой из очередей осуществляется параллельно и независимо от подсчета вероятностей выбора других очередей. Однако, в соответствии с предлагаемым способом, происходит подсчет сумм вероятностей (кумулятивных вероятностей). Этот подсчет также ведется в узле подсчета вероятностей выбора очередей на обслуживание.

Узел хранения постоянных величин управляющего блока содержит некоторые числовые константы, предназначенные для расчетов вероятностей, такие как: выделяемые доли обслуживания под каждый тип трафика, приоритеты по умолчанию (начальные значения вероятностей), семя генератора псевдослучайных чисел.

Генератор псевдослучайных чисел в процессе инициализации получает семя из узла хранения постоянных величин, порождает случайное число через фиксированное число тактов часов так, чтобы на каждый цикл диспетчеризации приходилось не менее одного вычисление псевдослучайного числа. Порожденная последовательность псевдослучайных чисел подчиняется равномерному распределению. Любое число из этой последовательности обладает неотрицательным значением, меньшим единицы.

Узел сравнения величин сравнивает значения, подсчитанные узлом подсчета вероятностей выбора очереди на обслуживание, т.е. кумулятивные вероятности, и генератором случайной величины. Для каждого типа трафика ставится либо значение истины или значение лжи в зависимости от разности значений.

Узел подсчета номера обслуживаемой очереди на основе данных сравнения, полученных от узла сравнения величин, производит выдачу обрабатывающим узлам, схеме выборки кадра обрабатывающего блока, а также узлу надзора сигналов разрешения или запрета обслуживания по каждому из классов обслуживания в соответствии с алгоритмом, представленным на рис. 1.

Приведенный вариант реализации стохастического алгоритма может быть применен не только для диспетчеризации очередей в коммутаторах, но и в других телекоммуникационных устройствах второго и третьего уровней сетевой модели.

Библиографический список

1. Олифер, В. Г. Компьютерные сети. Принципы, технологии, протоколы / В. Г. Олифер, Н. А. Олифер. – 4-е изд. – СПб. : Питер, 2010. – С. 943.

2. Семенов, А. О. Формальное представление стохастического алгоритма диспетчеризации очередей в телекоммуникационном оборудовании / А. О. Семенов, Н. Н. Коннов // Новые информационные технологии и системы (НИТиС-2014) : сб. науч. ст. XI Междунар. науч.-практ. конф. (г. Пенза, 25–27 ноября 2014 г.). – Пенза : Изд-во ПГУ, 2014. – С. 301–306.

3. Вегешна, Ш. Качество обслуживания в сетях IP / Ш. Вегешна. – М. : Вильямс, 2003. – С. 368.

4. Кучерявый, Е. А. Управление трафиком и качество обслуживания в сети Интернет / Е. А. Кучерявый. – СПб. : Наука и Техника, 2004. – С. 336.

А. О. Семенов, Н. Н. Коннов

Пензенский государственный университет, Пенза, Россия

МОДЕЛИРОВАНИЕ АЛГОРИТМА СТОХАСТИЧЕСКОЙ ДИСПЕТЧЕРИЗАЦИИ ОЧЕРЕДЕЙ В ПАКЕТЕ

Приводится описание особенностей некоторых временных характеристик модели алгоритма стохастической диспетчеризации очередей в составе сетевых коммутаторов с использованием сетей Петри.

В настоящей работе предлагаются временные характеристики сетей Петри, реализующих различные модификации стохастического алгоритма диспетчеризации, основанного на стохастическом подходе к выбору очередей на обслуживание с адаптацией к длинам очередей и текущему трафику в отдельных классах QoS [1– 4].

В данной работе рассматриваются различные временные характеристики для трѐх алгоритмов: дефицитный циклический (DRR), стохастический с возвратом (St1), стохастический без возврата (St2) [5].

Эффективность алгоритмов была исследована средствами инструментария цветных сетей Петри в программном пакете CPN Tools. В качестве входных данных на входы модели коммутаторов, содержащих три различных исполнения устройств диспетчеризации, подавались два различных аргумента: трафик, согласованный с выделенными полосами, выделенными под каждую совокупность классов обслуживания; и трафик, несогласованный с выделенными полосами, выделенными под каждую совокупность классов обслуживания. В рамках данной работы исследовалась только зависимость дисциплины выбора следующей очереди на обслуживание. При этом, количество выбираемых за одну выборку дисциплиной обслуживания бит не рассматривалось и имело квотный характер, как в алгоритме DRR. Отношения квот соответствуют k_1 / k_2/ k_3/ k_4=4/3/2/1. Выборка каждой очереди, при этом, равновероятна (p_1/p_2/p_3/p_4 = 1/1/1/1).

В качестве исследуемых величин использовались средняя задержка за всѐ время моделирования по всем классам обслуживания (см. строку «Ср задержка» в табл. 1); средняя задержка по каждой совокупности классов обслуживания (0-1; 2 3; 4 5; 6 7); размах средних задержек по каждому классу обслуживания (Размах); максимальная задержка (Макс задержка); максимальная задержка по каждому классу обслуживания (0 1; 2 3; 4 5; 6 7); размах максимальной задержки по классам обслуживания (Размах макс задержки); стандартное отклонение задержки по всем классам обслуживания (Стд откл); стандартное отклонение по каждому классу обслуживания (0 1; 2 3; 4 5; 6 7); размах стандартного отклонения по классам обслуживания (Размах стд откл); а также не количественные, а относительные показатели разброса величины задержки кадров коммутатором, а именно: отношение значения стандартного отклонения к среднему значению задержки кадров по всем классам обслуживания (Отношение откл к ср зад); отношение значения стандартного отклонения к среднему значению задержки кадров по каждой совокупности классов обслуживания (0 1; 2 3; 4 5; 6 7); размах отношений значений стандартного отклонения к среднему значению задержки кадров по всем классам обслуживания (Размах отн).

В результате исследования выяснилось, что стохастические алгоритмы показывают лучшие в сравнении с циклическим статистические и временные характеристики при моделировании, использующем в качестве аргументов данные несогласованного с выделенными полосами трафика (столбец «Несогласованный трафик»). Это касается,

в частности, таких величин, как размах средней задержки, максимальная задержка, размах максимальной задержки (преимущество почти в 5 раз), размах стандартного отклонения и др. (табл. 1).

Таблица 1

Временные характеристики различных алгоритмов

	Согласованный трафик			Несогласованный трафик		
CoS	DRR	ST1	ST2	DRR	ST1	ST2
Ср задержка	24744	24744	24744	25020	25020	25020
0-1	25256	31181	30700	21255	24325	25026
2 3	23987	24299	25500	21162	24662	27748
4 5	26077	24069	20986	22498	27352	27132
6 7	22353	15766	11557	34905	24987	25104
Размах	3723	15415	19142	13742	3026	2723
Макс задержка	225852	256705	217916	418825	241215	216241
0 1	122521	173082	132154	141439	24326	158218
2 3	176252	256705	197346	96584	24662	176172
4 5	170288	212125	217916	159888	27352	206364
6 7	225852	242543	188333	415825	24987	216241
Размах макс задержки	103331	83623	85762	322241	82645	58023
Стд откл	28104	24128	31475	31500	32617	32398
0 1	25754	39905	34444	20783	30154	29598
2 3	25332	28835	31818	20965	29203	32786
4 5	35473	33508	28508	25991	38006	34560
6 7	28682	18465	12974	47452	32087	32266
Размах стд откл	10141	21440	21470	26669	8803	4962
Отношение откл к ср зад	1,13579	0,975105	1,272026	1,258993	1,303637	1,294884
0 1	1,019718	1,279786	1,121954	0,977793	1,23963	1,18269
2 3	1,056072	1,186674	1,247765	0,990691	1,184129	1,181563
4 5	1,360318	1,392164	1,358429	1,155258	1,389514	1,273773
6 7	1,283139	1,171191	1,12261	1,359461	1,284148	1,285293
Размах отн	0,340599	0,220973	0,236475	0,381668	0,205385	0,103731

Что касается моделирования с использованием согласованного с выделяемыми полосами трафика, то по величине максимальной задержки, от которой косвенно зависит доля потерянных данных, преимущество зафиксировано только у алгоритма St2, в то время как по значению размаха этой величины преимущество имеют оба алгоритма стохастического толка. Однако по значению размаха средней задержки преимущество, всѐ же, отмечено у циклического алгоритма.

Таким образом, приведены результаты моделирования стохастического алгоритма. Приведены временные характеристики различных алгоритмов при моделировании на различных входных данных, но одинаковых порциях выборки.

Библиографический список

1. Олифер, В. Г. Компьютерные сети. Принципы, технологии, протоколы / В. Г. Олифер, Н. А. Олифер. – 4-е изд. – СПб. : Питер, 2010. – С. 943.

2. Семенов, А. О. Формальное представление стохастического алгоритма диспетчеризации очередей в телекоммуникационном оборудовании / А. О. Семенов, Н. Н. Коннов // Новые информационные технологии и системы (НИТиС-2014) : сб. науч. ст. XI Междунар. науч.-практ. конф. (г. Пенза, 25–27 ноября 2014 г.) . – Пенза : Изд-во ПГУ, 2014. – С. 301–306.

3. Вегешна, Ш. Качество обслуживания в сетях IP / Ш. Вегешна. – М. : Вильямс, 2003. – С. 368.
4. Кучерявый, Е. А. Управление трафиком и качество обслуживания в сети Интернет / Е. А. Кучерявый. – СПб. : Наука и Техника, 2004. – С. 336.
5. Феллер, Д. Теория вероятностей и еѐ приложения / Д. Феллер. – М. : Финансы и статистика, 1981. – С. 458.

Ю. Ю. Аладьев, К. И. Никишин

Пензенский государственный университет, Пенза, Россия

МОДЕЛИРОВАНИЕ СЕТЕВОГО ТРАФИКА В ПАКЕТЕ CPN TOOLs

Описана модель на основе сетей Петри генератора сетевого трафика для моделей сетей с поддержкой качества обслуживания. Подробно описан формат кадра, основные функции работы модели данного генератора трафика. Для построения модели использовался пакет CPN Tools. Данные, полученные в результате моделирования разработанного генератора, показали, что он может быть использован при исследовании и разработки новых механизмов поддержки качества обслуживания.

Генерация сетевого трафика с заданными характеристиками является важной задачей моделирования компьютерных сетей определенного назначения в частности, с поддержкой качества обслуживания (QoS). Моделирование обработки сетевого трафика коммутатором Ethernet позволяет оценить эффективность обработки трафика путем оценки вероятностных характеристик, в частности джиттера. Перспективным направлением моделирования телекоммуникационных сетей является применение математического аппарата иерархических временных цветных сетей Петри, поддерживаемого свободно распространяемым пакетом CPN Tools, который позволяет подробно исследовать различные вопросы поведения сложных компьютерных сетей. В CPN Tools язык описания моделей представляет собой сочетание графа сети Петри и языка программирования CPN ML [1, 2].

Обычно генерация трафика для моделей реализуется специальной сетью Петри. В частности известны работы, позволяющие генерировать трафик с заданными законами распределения длительности интервалов между поступлениями кадров, длительностью кадров [3, 4].

В настоящей работе представляется генератор сетевого трафика для моделей сетей с поддержкой QoS, в которых обеспечивается возможность дополнительно задавать законы распределения вероятности появления кадров различных классов QoS.

Модель генератора сетевого трафика на основе сетей Петри в пакете CPN Tools с выделением подсетей представлена на рис. 1 и состоит из следующих блоков:

- генератор стохастического трафика;
- генератор регулярного трафика;
- смеситель кадров трафика;
- блок записи полученной последовательности кадров в файл.

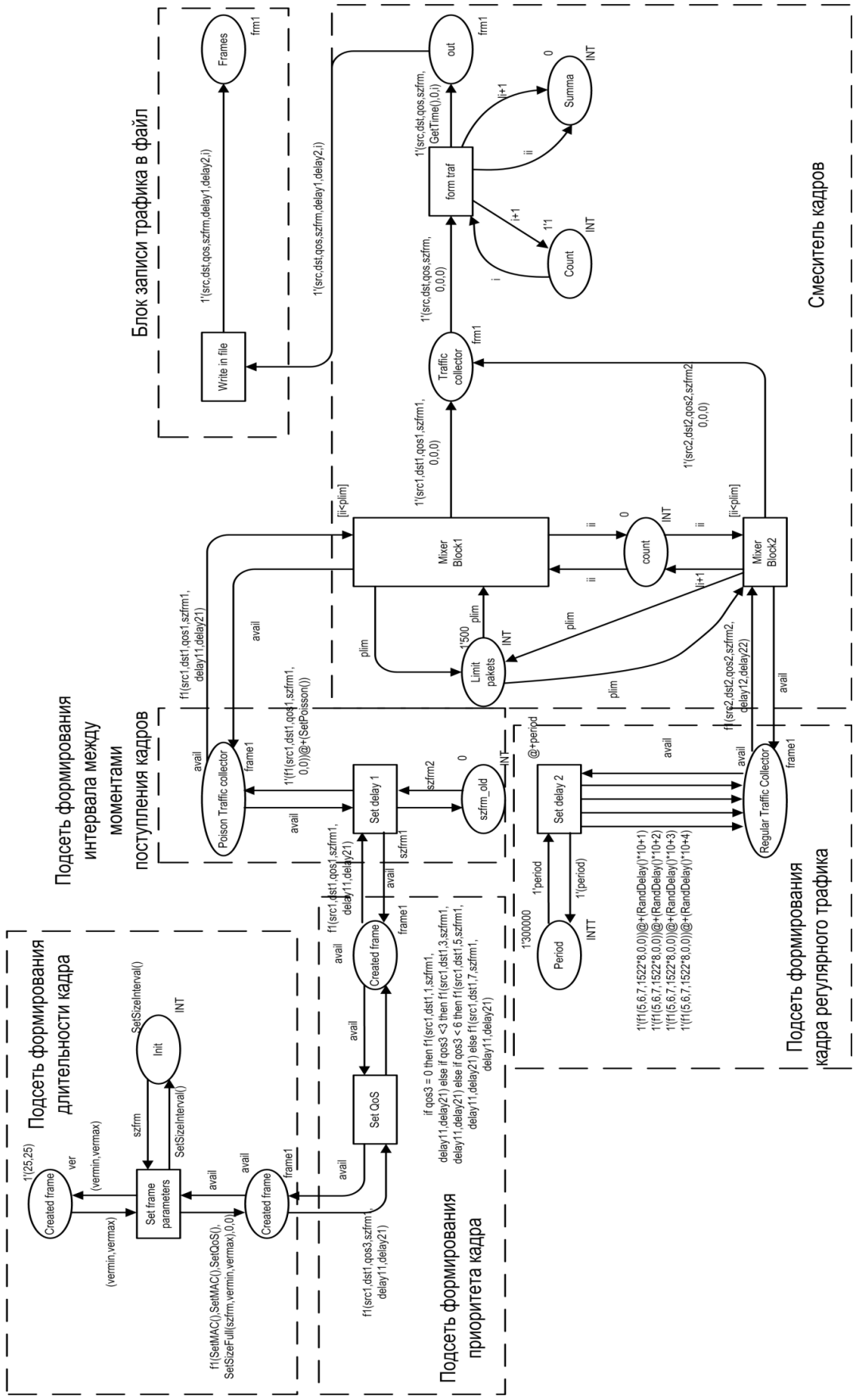


Рис. 1. Модель сети Петри генерирующий сетевой трафик

Генераторы регулярного и стохастического трафиков вырабатывают последовательности маркеров, соответствующие кадрам трафика. Смеситель кадров обеспечивает смешивание отдельных потоков кадров различных видов трафика (стохастического и регулярного) и контролирует длину их генерируемого потока. С помощью блока записи полученная на выходе последовательность кадров записывается в файл.

В модели генератора сетевого трафика представлены следующие цвета:

- цвет frm, который описывает структуру кадра и состоит из MAC-адресов источника и приемника, приоритета кадра, размера передаваемых данных и двух полей задержек;
- цвет frame, который может быть либо занят передачей кадра (frm), либо свободен (avail);
- цвет QoS, который описывает приоритет кадра;
- цвет MAC, который описывает адрес источника или приемника кадра;
- цвет INTT, который описывает временные задержки;
- цвет FRM_SIZE, который описывает размер кадра.

Все параметры кадра являются настраиваемыми, за счет этого обеспечивается гибкость систем.

В модели MAC-адрес источника или приемника кадра определяется функцией *fun SetMAC() = MAC.ran()* в диапазоне от 1 до 10.

Другим важным параметром кадра является приоритет кадра. В данной модели использовался частный случай с четырьмя видами приоритетов: от высокого приоритета к самому низкому с вероятностью распределения 40:30:20:10 процентов. Высокий приоритет имеет всегда значение 7, низкий имеет значение 1. Приоритет регулярного трафика всегда высокий. Приоритет стохастического трафика генерируется в два этапа. Сначала формируются значения по равномерному распределению в диапазоне от 0 до 9 *(fun SetQoS() = = discrete(0,9))* и в зависимости от интервала попадания формируется итоговый приоритет кадра. В результате получается, что приоритеты стохастического трафика распределены с вероятностью 40:30:20:10 от наиболее высокого приоритета к самому низкому. Ниже приведен фрагмент определения приоритета кадра.

```
qos3 = discrete(0,9).
if qos3 = 0 then f1(src1,dst1,1,szfrm1,delay11,delay21)
else if qos3 <3 then f1(src1,dst1,3,szfrm1,delay11,delay21)
else if qos3 < 6 then f1(src1,dst1,5,szfrm1,delay11,delay21)
else f1(src1,dst1,7,szfrm1,delay11,delay21)
```

Следующий параметр это длина кадра. Распределение длины поля данных имеет ярко выраженный бимодальный характер: до 50 % всех кадров имеют минимальную или максимальную длину (64 и 1500 байт, соответственно), длину остальных можно считать равномерно распределенной в оставшемся диапазоне[5].

Формирование кадров стохастического трафика происходило через временные интервалы, распределенные по экспоненциальному закону. Через экспоненциальное распределение в функции *fun SetPoisson() = floor(exponential(1.0/10000.0)+0.5)* можно регулировать загрузку канала.

Формирование кадров регулярного трафика происходило с определенным постоянным периодом. Одновременно формировались 4 кадра. Для устранения одновременного прихода кадров добавлялась задержка, которая вычислялась по формуле:

$$\text{Задержка} = discrete(1,4) * 10 + \text{Номер\_кадра}$$

Запись полученного трафика в файл осуществлялась стандартными средствами пакета CPN Tools с помощью монитора. Фиксация сгенерированного трафика в файле обеспечивает повторяемость экспериментов при смене или коррекции исследуемой модели компьютерной сети, тем самым позволяет анализировать и выбирать наилучшую модель в конкретной ситуации.

На рис. 2 представлены диаграммы вероятностных характеристик трафика, полученные в результате моделирования разработанного генератора сетевого трафика.

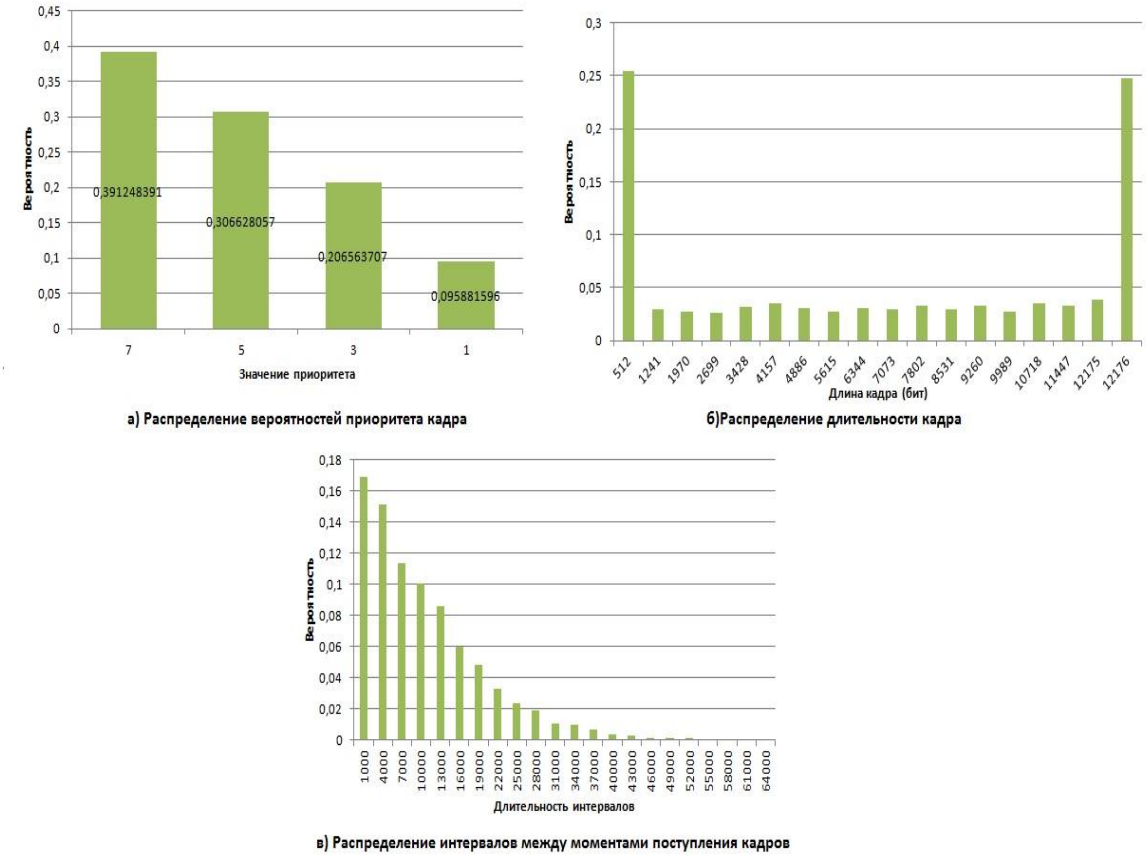


Рис. 2. Результаты моделирования

Данные, полученные в результате экспериментального исследования работы предлагаемого генератора показали, что он может быть использован при исследовании методами имитационного моделирования компьютерных сетей с поддержкой QoS с целью параметрической оптимизации работы корпоративных сетей или разработки новых механизмов поддержки качества обслуживания.

Библиографический список

1. Jensen, K. Coloured Petri Nets: modelling and validation of concurrent systems / Kurt Jensen, Lars M. Kristensen. – Springer, 2009. – 384 p.

2. Зайцев, Д. А. Моделирование телекоммуникационных систем в CPNTools / Д. А. Зайцев, Т. Р. Шмелева. – Одесса : Одесская национальная академия связи им. А. С. Попова, 2008. – 68 с.

3. Никишин, К. И. Моделирование трафика сети Ethernet цветными сетями Петри / К. И. Никишин, Н. Н. Коннов, А. Л. Домнин // Современные проблемы компьютерных наук (СПКН-2013) : сб. материалов I Междунар. науч.-практ. конф., посвящ. 70-летию образования Пензенского государственного университета (г. Пенза, 29–30 октября 2013 г.) / под ред. Л. Р. Фионовой. – Пенза : Изд-во ПГУ, 2013 – С. 120–124.

4. Никишин, К. И. Генератор трафика ethernet на основе цветных сетей петри / К. И. Никишин, Н. Н. Коннов // Модели, системы, сети в экономике, технике, природе и обществе. – 2016. – № 1(17). – С. 299–307.

5. Петров, В. В. Структура телетрафика и алгоритм обеспечения качества обслуживания при влиянии эффекта самоподобия : автореф. дис. … канд. техн. наук : 05.12.13 / Петров В. В. – М., 2004. – 198 с.

М. Ю. Новиков

Пензенский государственный университет, Пенза, Россия

РАДИОСЕТЬ ПЕРЕДАЧИ ДАННЫХ SKYMAN/CA

Педставлен обзор продукта компании CompTEK AquA, коммуникационной структуры Sky Metropolitan Area, которая является разработкой группы российских инженеров.

Развитие информационных связей в крупных старых городах России в связи с резким возрастанием требований по производительности связной инфраструктуры из-за вовлечения в информационный обмен цифровых компьютерных потоков, зашло в тупик. Единственной связной инфраструктурой общего пользования является телефонная сеть. В США и других развитых странах телефонные сети строятся на современных цифровых принципах, что позволяет использовать их для обмена цифровой информацией с достаточной скоростью. Доставшаяся нам в наследство телефонная сеть даже при применении самых современных систем и методов сжатия информации в лучшем случае обеспечивает скорость около 28Кбит/с. Учитывая, что такая скорость достигается только в определенных непродолжительных благоприятных ситуациях, в большинстве случаев такой производительности уже явно недостаточно.

Все существующие и нарождающиеся компьютерные российские глобальные сети строятся на телефонных сетях или на соответствующих им по скорости, но несколько превосходящим по надежности ведомственных линиях связи. Это создает серьезные трудности в создании полноценных информационных связей между разрозненными локальными быстродействующими кабельными сетями предприятий и учреждений, ограниченных их территориями, узлами спутниковой и международных связей и отдельными потребителями информации. Идея создания новой всепокрывающей связной информационной инфраструктуры с использованием перспективных оптоволоконных сетей для соединения уже сегодня большого числа абонентов на обширных территориях трудно реализуема из-за колоссальных временных и финансовых затрат, что диктует необходимость параллельного создания альтернативных способов передачи информации.

Использование лазерных систем связи в открытом пространстве над городами средней полосы России не представляется возможным из-за погодных условий. Целью реализации данного проекта является создание беспроводной компьютерной сети общего пользования. Использование сверхвысокочастотного диапазона радиоволн в качестве среды передачи информации решает вопрос создания высокоскоростных компьютерных коммуникаций между абонентами практически независимо от их расположения в пределах выделенных территорий, быстро и без значительных капитальных затрат для абонента.

В основу архитектуры системы заложены следующие принципы:

- строгое соблюдение международных стандартов в области систем связи и информационно-вычислительных сетей;
- использование сертифицированной Министерством Связи Российской Федерации аппаратуры;
- максимальная прозрачность в использовании;
- совместимость с большей частью сетевого программного обеспечения;
- использование новейших достижений в информатике и связи с целью реализации всех необходимых прикладных функций и потребностей пользователей системы;
- ремонтопригодность системы, обеспечиваемая возможностью оперативной замены типовых блоков;
- живучесть сети, определяемой наличием резервных путей и гибкой, динамической маршрутизацией, а также целым комплексом других мер, обеспечивающих сохранение работоспособности системы в целом при неисправности отдельных ее элементов;

• управляемость системы, достигаемая с помощью полного дистанционного управления маршрутизаторами.

Ключевой концепцией данного проекта является использование новой технологии построения беспроводных компьютерных сетей, основанной на передаче шумоподобных сигналов. Данная технология позволяет без использования проводных каналов построить многосотовую структуру для крупных городов области и односотовые кусты в мелких населенных пунктах, не имеющих своей проводной инфраструктуры, необходимой для решения нарождающихся информационных задач. Эта технология дает возможность развернуть предлагаемую систему в течение считанных месяцев, обеспечив при этом беспрецедентную простоту и оперативность подключения конечных пользователей.

SkyMAN/CA базируется на технологии передачи на размытом спектре прямой последовательности (DSSS), что обеспечивает высокую помехозащищенность в полосе приема. Другим положительным моментом систем с DSSS является относительно несложная процедура получения разрешения на эксплуатацию системы. В SkyMAN/CA используются радиоустройства, совместимые в рамках стандартовIEEE 802.11/802.11b.

В беспроводных технологиях, ориентированных на множественный доступ особенно болезненным является вопрос об эффективном разделении эфира между станциями. Наиболее часто используемым в этих случаях является протокол CSMA/CA, который построен на базе широко распространенного протокола CSMA/CD, входящего в состав Ethernet. Однако, в отличие от Ethernet, он не обеспечивает прослушивание среды во время передачи, что приводит к чрезвычайно большим накладным расходам при множественном доступе. При количестве станций в сегменте около 25 и интенсивном трафике, реальная пропускная способность составляет около 18 % от максимальной. Технология SkyMAN/CA включает в себя поллинговые протоколы обслуживания среды, оптимизированные для обеспечения множественного доступа в соте(WMA, WMA-CD).

За взаимодействие элементов SkyMAN/CA отвечает протокол множественного доступа, основанный на маршрутизации. RMA (Routed Multiple Access protocol), то есть протокол множественного доступа, основанный на маршрутизации, является одним из главных элементов архитектуры SkyMAN/CA. RMA позволяет в полной мере использовать преимущества сотовой архитектуры с применением многосекторных базовых станций. Протокол RMA построен на 3 основных механизмах:

• протокол тестирования каналов;

• регистрация абонентов на базовой станции с постоянным контролем качества канала;

• механизм роуминга (перемещение абонента от станции к станции) основанный на динамической IP маршрутизации.

На каждой базовой станции в таблицах хранится информация о правах и параметрах всех абонентов, которые могут быть к ней подключены. При разрастании сети объем информации возрастает, что приводит к соответствующим затруднениям в работе оператора сети.

Служба разрешения удаленного доступа RAPSэффективно решает эту проблему путем централизации всей информации об абонентских блоках в одной базе данных и автоматизации процесса конфигурации. Практически отпадает всякая необходимость непосредственного управления базовыми станциями, которые выполняют роль посредника между абонентским блоком и сервером RAPS.

SkyMAN/CA использует метод передачи информации с коммутацией пакетов. Базовым сетевым протоколом в системе является протоколIP (Internetwork Protocol), применяемый в сети Internet. Все остальные сетевые протоколы (такие, как IPX, Netbios, HDLC, SDLC, CLNS, X.25) должны инкапсулироваться в протокол IP устройствами пользователя. Этот подход позволяет использовать главные преимущества протокола IP, такие, как эффективная многопутевая маршрутизация, маршрутизация с объявлением состояния о канале (link-state), протоколRMA(Routed Multiple Access Protocol) и многие др.

Главной особенностью услуг, предоставляемых сетью SkyMAN, является то, что в отличие от других сетей передачи данных, использующих коммутируемые телефонные или выделенные линии связи на арендной основе, сеть SkyMAN имеет свою собственную транспортную структуру, специализированную на передаче цифровой информации, обеспечивающую для конечного пользователя скоростные и надежностные характеристики, недостижимые в других сетях при соизмеримых затратах.

Сеть реализуется в виде сотовой структуры, образуемой базовыми станциями (БС). К сети могут подключаться абоненты, расположенные в прямой видимости одной из БС. БС соединяются между собой и основными информационными узлами города беспроводной магистралью. Базовые станции размешаются на главенствующих высотах города, что дает возможность реализовать максимальное количество запросов на подключение. С точки зрения структуры, сеть SkyMAN представляет собой множество сетей, объединенных специализированными радиомаршрутизаторами Callisto, поддерживающими протоколы RTIP, TCP/IP, IPX/SPX, X.25. Использование множества взаимодействующих маршрутизаторов с централизованным администрированием позволяет обеспечить полный контроль за трафиком в сети. Многоуровневая система защиты от несанкционированного доступа предотвращает бесконтрольный доступ к сетям абонентов извне.

Сеть пользователя стыкуется с Сетью при помощи блоков доступа к сети, являющимися техническими границами Сети. Блок доступа к SkyMAN является радиомаршрутизатором. Программное обеспечение блоков доступа обеспечивает изоляцию трафика Сети от трафика сети пользователя и их взаимодействие в соответствии с параметрами, устанавливаемыми дистанционно с центра управления сетью. Протоколы взаимодействия с сетями пользователей: TCP/IP, IPX, X.25, RTIP. Антенно-кабельная система (от комнатной антенны, при наличии прямой видимости БС из окна здания абонента, до внешней параболической антенны) обеспечивает соединение блока доступа с БС.

Скоростные характеристики блока доступа.

Максимальная программная скорость передачи данных при связи с другим абонентом внутри одной соты – БС – 400 Кбит/с;

Максимальная программная скорость передачи данных при связи с другим абонентом другой соты – БС – 800 Кбит/с;

Блок доступа можно подключать к локальной сети абонента по оптоволокну, обеспечивая этим гальваническую развязку с локальной сетью и снимая ограничение на расстояние между локальной сетью и местом установки блока доступа. Дальность от центра соты до абонента зависит от комплектации базовой станции и абонента. Для обеспечения максимальной производительности системы необходимо не допускать работу линии без запаса по энергетике (работу на максимальной дальности). Для работы в любую погоду необходим запас на снижение эффективности антенн из-за осадков в виде мокрого снега. Затуханием сигнала на трассе из-за атмосферных осадков можно пренебречь из-за незначительного поглощения сигналов частоты 2400 МГц.

Питание БС осуществляется через непрерываемые источники питания типа APC Smart UPS 700XL, обеспечивающие мониторинг наличия напряжения сети, передачу информации в центр управления сетью, сохранение энергопитания БС при отключении внешнего энергопитания на время более 10 ч.

Управление сетью обеспечивается с Центра управления сетью (ЦУС). Мащрутизатор каждой из БС обеспечивает мониторинг соединенных с ним сетей и предоставляет необходимую информацию ЦУС.

Сопряжение сети с сетями Internet, INFOTEL и другими глобальными сетями обеспечивается центром коммутации пакетов. Данное решение обеспечивать для своих пользователей техническую возможность обмена данными с пользователями любой другой СДОП, либо с использованием непосредственного межсетевого сопряжения, либо с помощью транзита через третьи сети. Реализация такой возможности осуществляется по соглашению между операторами сетей.

Библиографический список

1. Кучерявый, Е. А. Управление трафиком и качество обслуживания в сети Интернет / Е. А. Кучерявый. – СПб. : Наука и Техника, 2004. – С. 336.
2. Мешковский, К. А. Цифровые системы передачи: принципы работы / К. А. Мешковский. – СПб. : Энергомашиностроение, 2007.

Д. А. Замятин, В. С. Богданова, С. В. Степанова

Сибирский государственный университет науки и технологий
имени академика М. Ф. Решетнева (СибГУ), Красноярск, Россия

КОНЦЕПЦИЯ НЕЗАВИСИМОЙ СПУТНИКОВОЙ КОМПЬЮТЕРНОЙ СЕТИ

Описывается разработка концепции крупномасштабной компьютерной сети с размещением центров обмена данными на орбите Земли и независимой от сети Интернет.

На сегодняшний день одной из важных задач развития информационных технологий является повышение уровня сетевой безопасности. В частности, актуальна проблема нахождения компромисса между обеспечением с одной стороны эффективного противодействия экстремизму, а с другой стороны – возможности рядовых пользователей при пользовании сетью чувствовать себя достаточно комфортно, не опасаясь утечки персональных данных.

Принципы работы сети Интернет практически исключают такой компромисс, что делает необходимым создание независимой от Интернета компьютерной сети с совершенно иными принципами, частично имеющей сходство с компьютерными сетями военного назначения, что не позволит сети стать источником вреда как для ее пользователей, так и для общественной безопасности.

Основные принципы проекта сети:

1) деловая и научная направленность;
2) полная независимость от сети Интернет, на аппаратном уровне;
3) ограничение возможности внесения контента с клиентских устройств;
4) обеспечение максимально возможной конфиденциальности обмена информацией между пользователями.

В числе прочих особенностей проекта сети важное место занимает отсутствие проводной связи крупных серверных комплексов как с клиентскими устройствами, так и между собой. Вместо этого вся связь осуществляется через спутники связи и орбитальные серверные комплексы. На ранних этапах проекта рассматривался вариант размещения всех без исключения физических носителей информации такой сети на орбите Земли, что на сегодняшний день является весьма сложной и дорогостоящей для реализации идеей. В связи с этим было принято решение несколько скорректировать вектор развития проекта [1].

Таким образом, на сегодняшний день цель проекта – найти оптимальный вариант концепции крупной компьютерной сети, с полной независимостью от Интернета и повышенным уровнем защищенности.

Задачи:

1) оценка возможностей вывода на орбиту крупных серверных комплексов и формирования орбитальных центров обмена данными;

2) выбор целевой орбиты и подходящего носителя для заданной цели;

3) разработка концепции космического аппарата для решения поставленных задач;

4) решение вопроса обеспечения надежной связи между космическими аппаратами и с наземными устройствами.

Размещение аппаратного обеспечения крупной компьютерной сети на орбите Земли делает необходимым разработку крупномасштабной космической программы. В данной работе рассмотрена оптимизация изначальной концепции сети с учетом выявленных на предыдущих этапах преимуществ и недостатков.

Уязвимость к космическому мусору, сложность доставки на орбиту и сильно ограниченные возможности обслуживания на данный момент делают полное размещение всех вычислительных мощностей на космических аппаратах не вполне оптимальным вариантом. К первоначальной концепции полного размещения всех центров обмена данными в космосе планируется вернуться на более поздних этапах развития, после появления и широкого распространения новых поколений информационных носителей с более высокой плотностью данных, что сделает возможным создание серверов меньших размеров при той же вычислительной мощности.

Размещение центров обмена данными на орбите, однако, обеспечивает большую зону покрытия сети и высокую скорость передачи информации, причем даже в местности, где другие виды связи недоступны, а недосягаемость для человека может являться не минусом, а плюсом, ввиду высокой защищенности от злоумышленников. Поэтому на текущем этапе был предложен компромиссный вариант. Основные узлы сети – орбитальные модули связи, служащие для обмена информацией с наземными приемно-передающими устройствами и другими такими же КА. К модулю связи пристыковываются блоки хранения и обработки данных [1].

В отличие от первоначального варианта, как таковой серверный комплекс не размещается полностью на космическом аппарате по причине слишком большой массы, осложняющей задачу доставки КА на целевую орбиту и ограничивающей возможности обслуживания серверного комплекса. Часть комплекса находится на Земле, в космосе же предлагается оставить так называемые «кэшевые» серверы, так как на них хранятся лишь копии данных, находящихся непосредственно в работе, т.е. в течение сравнительно непродолжительного времени.

Следовательно, орбитальный центр обмена данными имеет сравнительно небольшую массу, что также обусловлено отсутствием крупных систем охлаждения серверов ввиду низкой температуры окружающей среды. Таким образом, размещение части аппаратного обеспечения сети на Земле решает проблемы как охлаждения серверных комплексов, так и их доставки на орбиту.

Общий принцип передачи данных при этом выглядит следующим образом: крупный наземный центр хранения информации обменивается данными с геостационарным узлом связи, расположенным на соответствующей долготе, а узел связи передает пакеты данных на низкоорбитальные кэшевые серверы, ближайшие к тому или иному клиентскому устройству, запрашивающему информацию.

Одним из определяющих параметров при выборе орбиты является масса космических аппаратов. Другим – скорость обмена данными между орбитальным дата-центром и наземными устройствами. При оценке этих параметров наиболее уместным решением выглядит размещение кэшевых орбитальных серверов на низкой околоземной орбите (НОО) и узлов связи на геостационарной орбите (ГСО) [2].

В качестве предполагаемого средства выведения серверных комплексов на орбиту рассматриваются российские ракеты-носители семейства «Ангара», что открывает достаточно широкие возможности при выборе оптимальной орбиты и проектировании космического аппарата для представленных целей [3].

Следующий этап работы включает в себя расчет массогабаритных характеристик космического аппарата для заданных целей и выбор наиболее подходящей для него орбиты, а также уточнение выбора носителя.

Таблица 1

Характеристики ракет семейства «Ангара»

	Ангара 1.2	Ангара А3	Ангара А5
Стартовая масса, т	171	481	773
Масса полезной нагрузки, т:			
– для НОО (Н = 200 км, i = 63°)	3,5	14,0	24,0
– на ГПО (Н = 5500 км, i = 25°)	–	2,4 («Бриз-М»)	5,4 («Бриз-М»)
			7,5 («КВТК»)
– на ГСО	–	1,0 («Бриз-М»)	2,8 («Бриз-М»)
			4,5 («КВТК»)

Исследование выполнено при поддержке краевого государственного автономного учреждения «Красноярский краевой фонд поддержки научной и научно-технической деятельности» в рамках участия в мероприятии XIV Международная научно-техническая конференция «Новые информационные технологии и системы (НИТиС-2017)».

Библиографический список

1. Замятин, Д. А. Разработка концепции орбитальной компьютерной сети / Д. А. Замятин, Е. П. Шашило, В. В. Кольга // Решетневские чтения : материалы XX Междунар. науч. конф. (г. Красноярск, 9–12 ноября 2016 г.) : в 2 ч. / под общ. ред. Ю. Ю. Логинова ; Сиб. гос. аэрокосм. ун-т. – Красноярск, 2016. – С. 27–28.
2. Наблюдение ИСЗ. – URL: http://www.sat.belastro.net/ (дата обращения: 14.09.2016).
3. ГКНПЦ им. М. В. Хруничева. Семейство ракет-носителей «Ангара». – URL: http://www.khrunichev.ru/main.php?id=44 (дата обращения: 20.05.2017).

М. Н. Коннов

Пензенский государственный университет, Пенза, Россия

ПОВЫШЕНИЕ ТОЧНОСТИ УСТАНОВКИ СЕТЕВОГО ВРЕМЕНИ ПРОТОКОЛОМ IEEE-1588

Рассматривается возможность повышения синхронизации по протоколу IEEE-1588 устройств в сети Ethernet за счет организации в коммутаторах виртуального тайслота.

Технология Ethernet в настоящее время начала широко применяться в системах промышленной автоматизации, в контрольно-измерительных и управляющих системах. Обязательным условием функционирования таких систем является синхронизация операций, выполняемых в объединенных сетью ЭВМ или контроллерах устройств. Для синхронизации часов реального времени в сетевых устройств используется протокол «точного времени» (PTP – Precision Time Protocol), который действует на втором уровне модели взаимодействия открытых систем (OSI). Протокол PTP описан в стандарте IEEE 1588 [1, 2].

Стандарт IEEE 1588 метод синхронизации устройств в сети с точностью выше 1 мкс (до 10 нс). Данный протокол обеспечивает синхронизацию ведомых устройств от ведущего, удостоверяясь, что события и временные метки на всех устройствах используют одну и ту же временную базу. В протоколе предусмотрены две ступени для синхронизации устройств: определение ведущего устройства и коррекция разбега во времени, вызванного смещением отсчета часов в каждом устройстве и задержками в передаче данных по сети.

При инициализации системы протокол PTP использует алгоритм «наилучших ведущих часов» для определения самого точного источника синхронизации в сети, который ведущим, а все остальные устройства в сети – ведомые и корректируют свои часы по ведущему устройству. Разница во времени между ведущим и ведомым устройствами является

комбинацией смещения отсчета часов и задержки передачи синхронизирующего сообщения. Поэтому коррекция временного сдвига должна выполняться в два этапа: вычисление задержек передачи и сдвига, а затем их коррекция.

Порядок синхронизации часов двух устройств приведен на рис. 1.

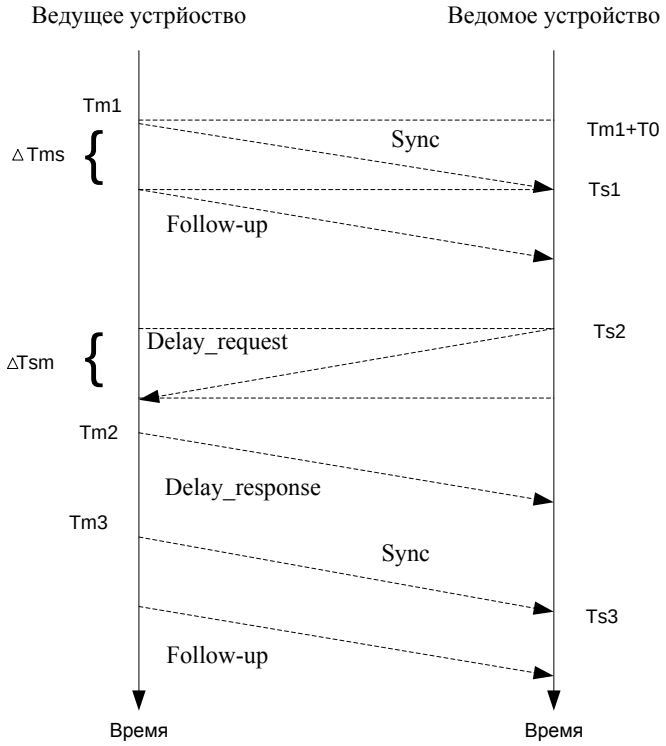


Рис. 1. Обмен сообщениями при синхронизации двух часов

В первой фазе корректируется расхождение между ведущими и ведомыми часами за счет циклической передачи ведущими часами сообщений, которые содержат расчетное значение точного времени его передачи (сообщения Sync и Follow-up). Подчиненные часы вычисляют коррекцию, учитывая при этом время передачи кадра, которое определяется во второй фаза процесса синхронизации.

С этой целью ведомые часы направляют ведущим пакет запроса Delay_request, а ведущие часы фиксируют время приема пакета и посылают его значение подчиненным часам в специальном пакете Delay_response. Затем подчиненные часы вычисляют время задержки, при этом особо важным является равнозначность задержки в каждом из направлений поскольку при вычислении задержки передачи сообщения принимается, что средняя задержка передачи данных в канале равна среднему арифметическому задержек распространения в разные стороны канала.

Окончательная синхронизация часов выполняется после отправки ведущим устройством второго набора сообщений Sync и Follow-up После этого ведомое устройство подстраивает свои часы в соответствии с вычисленными значениями.

Поскольку опорные источники синхронизации в каждом устройстве нестабильны, а задержки в канале могут меняться со временем, необходимо периодически повторять коррекцию часов ведомого устройства.

Точность синхронизации зависит от точности измерения задержек передачи кадра в каждом звене сети Ethernet, которая складывается из следующих составляющих:

- задержка распространения, вызванная конечной скоростью распространения сигнала в среде передачи (меди, оптоволокне);
- задержка сериализации, связанная с последовательной (побитной) передачей кадров в выходных портах;

• задержка коммутации, определяемая временем для продвижения кадра из входного в выходной порт;

• задержка в очереди, возникающая при задержке кадров из-за перегрузки сети.

Две последние составляющие носят случайный характер и зависят от нагрузки сети, причем задержка кадров в очередях может достигать сотен микросекунд.

В стандарте IEEE 1588 v.1 2002 г. подразумевается, что задержки в канале связи симметричны; то есть, задержка передачи в прямом канале равна задержке передачи в обратном канале. Кроме того, задержка не должна меняться в процессе ее измерения. Изменение задержки в процессе ее измерения приведет к асимметрии и джиттеру, которые неизбежно повлияют на точность синхронизации, поэтому протокол первой версии чрезвычайно чувствителен к разбросу задержек при вычислении коррекции.

В стандарте IEEE 1588 v.2 2008 г. При программно-аппаратной реализации наиболее чувствительные функции протокола, такие как запись временной метки PTP-пакета, реализуются на физическом уровне Ethernet, при прохождении пакета через коммутатор, позволяя добиться точности протокола PTP порядка 10–20 нс [3]. Естественно, вторая версия требует применение специальных сетевых плат и коммутаторов.

В работе [4] автор показал, что погрешности оценки задержек при использовании протокола PTP версии 2002 г. могут быть достаточно просто устранены за счет организации в коммутаторе передачи кадров через так называемый «виртуальный тайм-слот» (ВТС) с пост-блокировкой [5, 6].

Необходимыми условиями реализации ВТС являются:

• поддержка протоколов IEEE 802.1 q /p [7];

• кадрам PTP назначен наивысший приоритет;

• полоса, выделяемая в PTP-трафика, исключает накопление PTP кадров в буферах коммутаторов.

Передняя граница виртуального тайм-слота в каждом коммутаторе определяется моментом приема и распознания очередного PTP-кадра, после чего он задерживается в буфере высшего приоритета на фиксированное время Тбл > Тмах, где Тмах – время передачи кадра IEEE 802.1 q /p наибольшей длины. На временя блокировки запрещается начало передачи через выходной порт любых кадров. После снятия блокировки через освобожденный выходной порт начнется передача PTP-кадра и кадров данных.

Рассмотренный вариант управления трафиком позволяет исключить джиттер, но приводит к дополнительной фиксированной задержке PTP-кадра в каждом узле.

Однако дальнейшему увеличению точности препятствует собственный разброс задержек транспортировки кадров внутри коммутатора. Эта погрешность не может быть устранена организацией ВТС, описанной выше, так как момент блокировки связан с моментом классификации, выполняемым в порту назначения.

Автором предлагается способ устранения этой погрешности за счет привязки момента начала блокировки к моменту поступления или окончания приема пакета во входном порту. Цель – повышения точности протокола временной синхронизации в сети за счет фиксации задержек коммутатора по методу виртуального таймслота.

Естественно, что эти действия должны выполняться на физическом уровне, как показано на рис. 2.

Пусть в момент Т1 начнется прием очередного кадра в входной порт.

В момент Т2 прием закончится и по сигналу аппаратуры входного порта зафиксируется показание внетренних часов коммутатора (например, с точностью 10 нс). Код добавляется к принятому кадру и направляется в порт назначения.

В момент Т3 завершиться коммутация.

В момент Т4 произойдет идентификация типа кадра. Если кадр не относится к протоколу PTP. Он просто направляется в соответствующий буфер, а его метка времени отбрасывается. В противном случае метка будет в буфер атрибутов соответствующей высокоприоритетной очереди.

Пусть в момент Т5 выходной порт освободится, и схема обзора считает атрибут высокоприоритетной очереди. Наличие метки в этом буфере вызывает запуск блокировки ВТС.

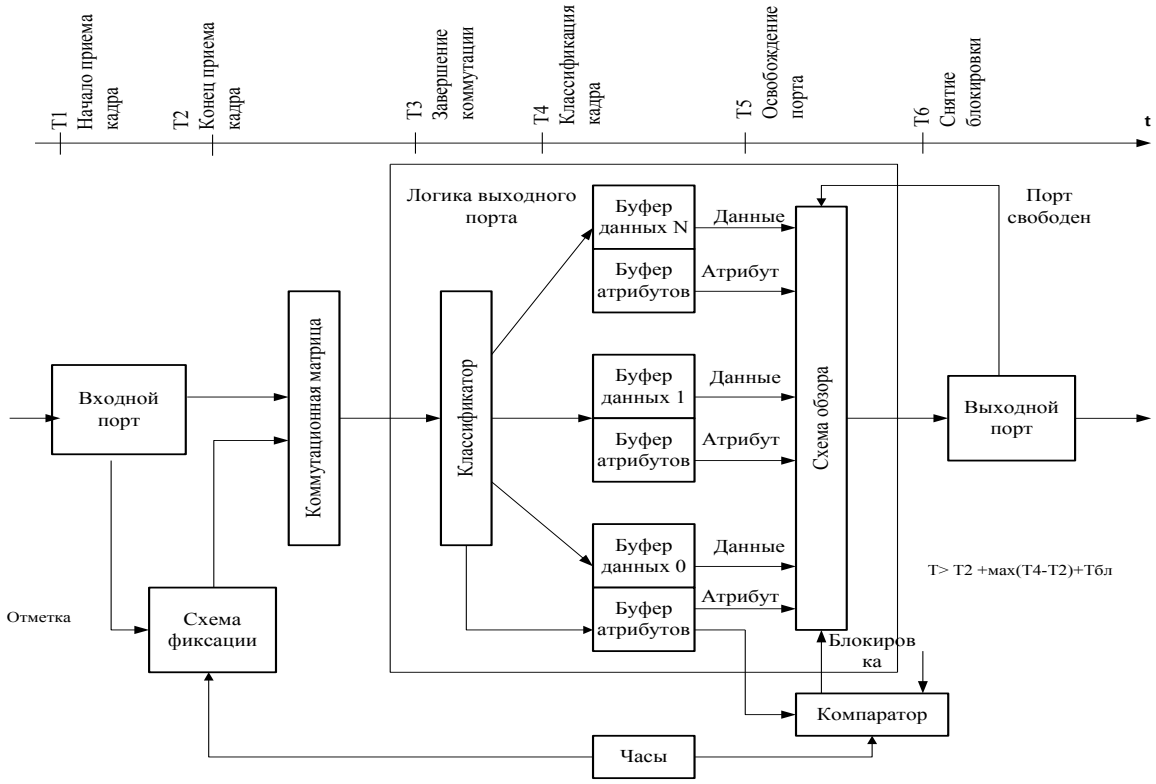


Рис. 2. Схема счет фиксации задержек коммутатора по методу виртуального таймслота

Момент снятия блокировки определится так $T > T2 + \max(T4–T2) + Тбл$, где $\max(T4–T2)$ – заранее заданный наибольший разброс задержки коммутации.

Пусть в момент T5 выходной порт освободится, и схема обзора считает из буфера атрибут высокоприоритетной очереди. Наличие метки PTP в этом буфере вызывает запуск блокировки BTC.

В момент T6 начинается передачи кадра через выходной порт. Временной интервал T6–T2 должен быть постоянен (с точностью < 10 нс). Метки времени в передаваемый кадр не помещаются.

Таким образом, повышения точности протокола временной синхронизации в сети будет достигнуто за счет фиксации задержек передачи PTP пакета через коммутатор по методу виртуального таймслота. Применение предложенного способа позволит упростить схему коммутатора, исключив трудоемкие операции по корректировки пакета PTP для записи в него измеренной задержки. Упрощается так же обработка кадра в ведомых часах и появляется возможность применять в овонечных устройствах обычные микроконтроллеры.

Библиографический список

1. Eidson, J. C. Measurement, Control, and Communication Using IEEE 1588 / J. C. Eidson. – Springer-Verlag London, 2006. – P. 284.

2. Телегин, С. Протокол PTP для синхронизации сетей NGN. Вопросы применения / С. Телегин // Первая миля. – 2009. – № 5-6. – С. 20–23.

3. Вербенин, Б. А. Особенности синхронизации протокола IEEE 1588 / Б. А. Вербенин // Электронные компоненты. – 2011. – № 11. – С. 52–55.

4. Коннов, М. Н. О повышении точности синхронизации сети Ethernet по протоколу IEEE 1588 / М. Н. Коннов // Информатика: проблемы, методология, технологии : сб. материалов XIV Междунар. науч.-метод. конф. : в 3 т. – Воронеж : Вэлборн, 2014. – Т. 3. – С. 210–214.

5. Коннов, М. Н. Об одном механизме передачи трафика реального времени в сетях с пакетной коммутацией / М. Н. Коннов, В. Б. Механов // Современные информационные технологии. – 2010. – № 11. – С. 139–141.

6. Механов, В. Б. Механизмы компенсации джиттера в коммутаторах Ethernet / В. Б. Механов, М. Н. Коннов // Телематика 2013 : тр. XX Всерос. науч.-метод. конф. – СПб. : СПбГУ ИТМО, 2013. – Т. 1. – С. 235–237.

7. Филимонов, А. Построение мультисервисных сетей Ethernet / А. Филимонов. – СПб. : БХВ-Петербург, 2007. – С. 592.

Е. А. Кизилов, Н. Н. Коннов, Д. В. Патунин, И. Ю. Фролов

Пензенский государственный университет, Пенза, Россия

ОЦЕНКА РАЗМЕРОВ ОЧЕРЕДЕЙ В КОММУТАТОРЕ С ПОДДЕРЖКОЙ QOS

Проводится сравнительный анализ статистических характеристик очередей в коммутаторах Ethernet с поддержкой качества обслуживания, использующих дефицитный циклический и динамический приоритетный алгоритмы диспетчеризации очередей.

В современных коммутаторах Ethernet функционируют службы поддержки качества обслуживания (QoS- quality of service), которые обеспечивают для трафика определенного класса заданные требования по полосе пропускания, задержке кадров, ее вариации (джиттера) и доле потерянных кадров. Важным механизмом службы QoS является диспетчеризация очередей в коммутаторах, от алгоритмов которой зависят значения задержки и джиттера передаваемых кадров, определяющие возможность передачи в сети Ethernet трафика реального времени [1].

Для диспетчеризации очередей широко используется циклические алгоритмы, в частности алгоритм дефицитного циклического обслуживания (DRR – Deficit Round Robin). В этом алгоритме для каждого класса QoS-трафика выделяется отдельная очередь. Очереди обслуживаются в порядке возрастания их номеров, причем при отсутствии кадров в некоторой очереди алгоритм переключается на другую очередь с большим номером. Количество данных, передаваемых из каждой очереди за один сеанс обслуживании, задается закрепленным за ней весом, пропорциональным полосе пропускания выходного порта, выделяемой для трафика соответствующего класса, и максимальному количеству (бит) байтов данных, передаваемых из очереди за один сеанс [2].

Достоинства DRR алгоритма – исключение монополизации полосы пропускания выходного порта, высокая точность задания выделяемой полосы пропускания для определенного класса трафика независимо от разброса размеров продвигаемых коммутатором кадров, а так же простота реализации. Недостатки DRR алгоритма - неконтролируемая величина джиттера коммутируемых кадров из-за разброса время ожидания в различных очередях, вызванного детерминистским порядком их обслуживания. Кроме того, задержка кадров определенного класса QoS, а так же размеры соответствующих очередей, зависят от выделенной для конкретного класса полосы пропускания, в результате алгоритмы не позволяют эффективно перераспределять неиспользуемую полосу пропускания недогруженных каналов для расширения полосы пропускания перегруженных. Это приводит при пульсирующем характере трафика или в случае несогласованности фактической нагрузки и выделенной полосы пропускания для трафика класса QoS к увеличении задержки кадров и перегрузке очередей в каналах с малой назначенной полосой пропускания.

Для решения этой проблемы был предложен динамический алгоритм диспетчеризации DRR-TSS (TSS –Time Selected Service), совмещающий принципы циклической диспетчеризации с учетом времени ожидания обслуживания кадров (временной селекцией) при ограничении длительности непрерывного обслуживания каждой очереди. Проведенные исследования джиттера кадров различного вида трафика показали, что DRR-TSS алгоритм обеспечивает более справедливость распределения выделяемой для каждого класса трафика полосы пропускания [3, 4].

В настоящей работе приводятся результаты исследований другой важной характеристики, показывающей эффективность алгоритма диспетчеризации очередей, – размеров очередей. Оценка эффективности алгоритмов динамической диспетчеризации по сравнению с алгоритмами DRR выполнялась их имитационным моделированием в пакете CPN Tools, использующем математический аппарат цветных иерархических временных сетей Петри [5]. Структура сетей Петри (рис. 1), моделирующих буфер выходного порта, в котором реализуются алгоритмы диспетчеризации очередей реализуются, были построены по принципам, изложенным в [3, 6, 7], при этом для контроля размеров очередей были в подсети Петри, моделирующие буферную память коммутатора, были добавлены специальные измерительные элементы.

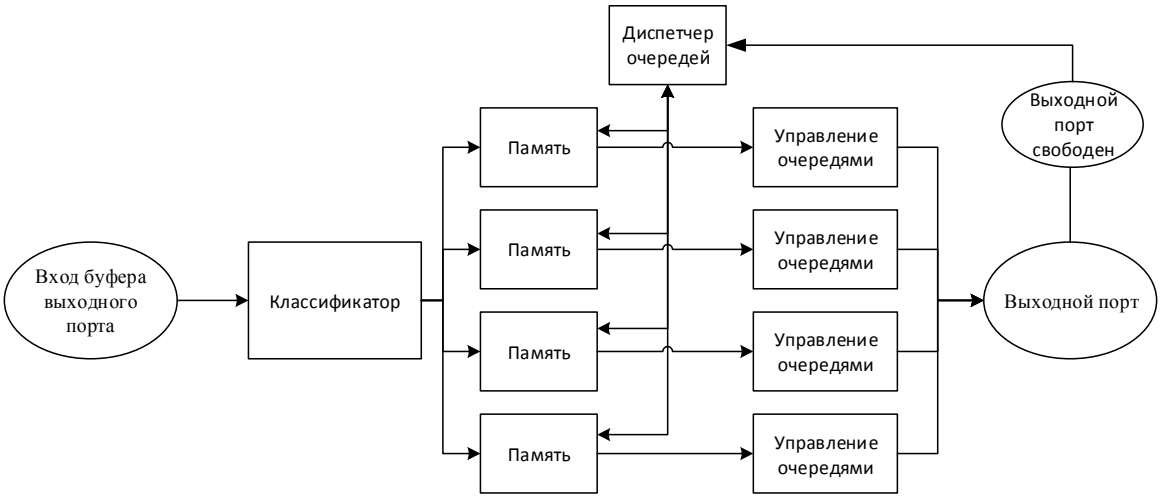


Рис. 1. Структура модели буфера выходного порта коммутатора

Пример такой подсети ***Queue*** – очереди типа FIFO, буферизирующей кадры одного класса, случая алгоритма DRR-TSS приведен на рис. 2. В этой же подсети также реализованы подсчет размера текущей очереди и отбрасывание поступивших кадров при переполнении буфера.

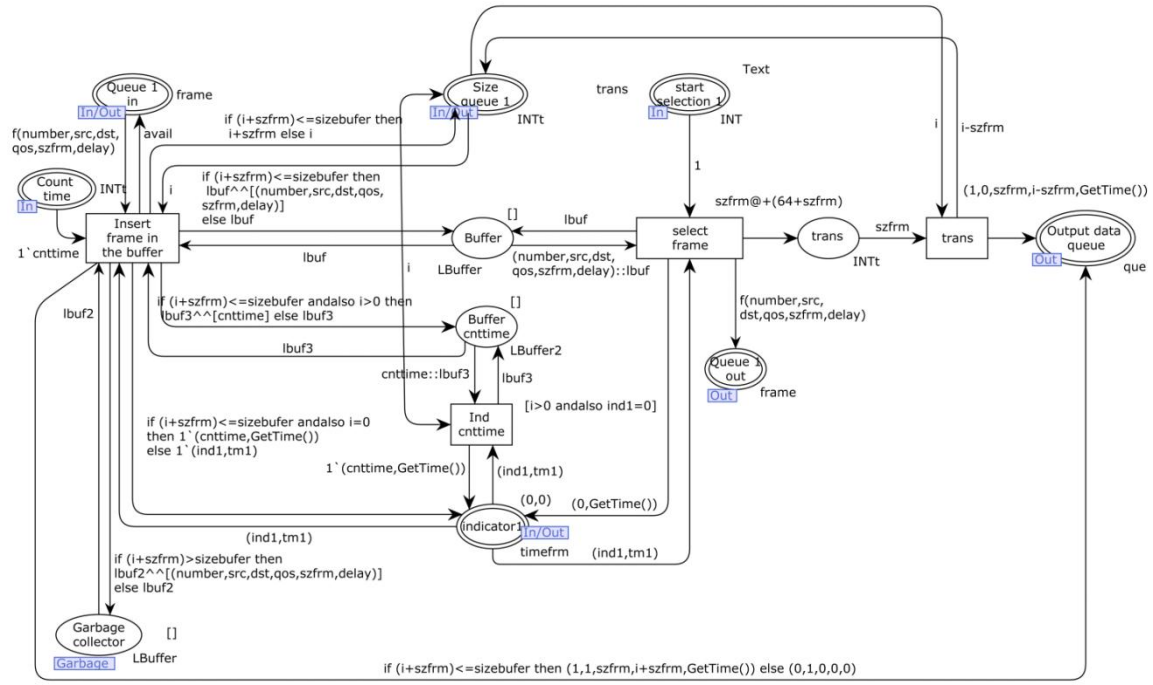


Рис. 2. Сеть Петри, моделирующая алгоритм работы памяти

При поступлении кадра в позицию ***Queue 1 in*** проверяется наличие свободного места в очереди. Если свободное место есть, то в позиции ***Size queue 1*** изменяется количество хранимых кадров на значение поля ***szfrm*** кадра (размер кадра в битах), в позицию ***Buffer*** в хвост очереди добавляется текущий кадр, Если в очереди нет свободного места, то кадр отбрасывается, помещаясь в позицию ***Garbadge collector.***

Когда в очередь поступает сигнал считывания кадра (позиция ***start selection 1***) из позиции ***Buffer*** считывается кадр, стоящий в голове очереди, и помещается в позицию ***Queue 1 out***.

Позиции ***Buffer cnttime*** и ***indicator1*** служат для формирования, хранения и выдачи специального атрибута каждого кадра, отмечающего его время поступления в очередь по временной метки позиции ***Count time***. Этот атрибут используется для временной селекцией кадров.

Позиции **Output data queue** и **Trans** совместно с одноименным переходом используются для сбора статистических данных о состоянии очереди, выполняемого следующим образом:

- при срабатывании перехода ***Insert frame in the buffer*** одновременно с помещением очередного кадра в буфер записывает в позицию ***Output data queue*** маркер с цветом ***1, 1, szfrm, i+ szfrm, getTime(),*** которая отмечает событие увеличения очереди 1 на ***szfrm*** бит в момент времени поступления кадра в очередь***;***
- при срабатывании перехода ***Select frame*** после чтения очередного кадра из буфера с задержкой ***64+ szfrm*** (время передачи кадра с учетом преамбулы [1]) в позицию **Trans** замисывается маркер, указывающий на размер переданного кадра;
- в позицию ***Output data queue*** перезод ***Trans*** записывает маркер с цветом ***1, 0, szfrm, i-szfrm, getTime(),*** которая отмечает событие уменьшения очереди 1 на величину ***szfrm*** бит в момент времени окончания передачи удаленного из очереди 1 кадра.

В результате работы модели в позиции ***Output data queue*** накапливаются маркеры, отмечающие динамику изменения очередей, последовательность значений цветов которых выводится в текстовый файл с помощью функции мониторинга, написанной на языке CPN ML. Дальнейшая обработка статистических данных очередей может выполняться любым математическим пакетом, например, Excel или MathCad.

Входной трафик моделировался с помощью специальной сети Петри [8] и представляет собой комбинацию случайного потока кадров с экспоненциальным распределением времени между кадрами и полимодальной длительностью кадров, и квазирегулярного потока одновременное поступающих кадров всех классов максимальной длинны имитирующих «пачечность». При этом варьировались такие параметры трафика, как общая интенсивность трафика, доли трафика различных классов QoS, имитируя потоки данных с параметрами трафика, которые согласованны (поток 1) и несогласованны (поток 2) с выделяемой полосой пропускания для различных классов QoS. представляет комбинацию потока кадров, принадлежащих с равной вероятностью 4 классам QoS (табл. 1).

Таблица 1

Распределение полосы пропускания между классами трафика для различных потоков

№ потока	Параметры	Классы *QoS:*			
		0–1	2–3	4–5	6–7
1	Выделяемая полоса пропускания, %	10	20	30	40
	Доля трафика	0,1	0,2	0,3	0,4
2	Выделяемая полоса пропускания, %	10	20	30	40
	Доля трафика	0,25	0,25	0,25	0,25

На рис. 2 представлен фрагмент Временной профиля изменения очереди 4 для различных алгоритмов.

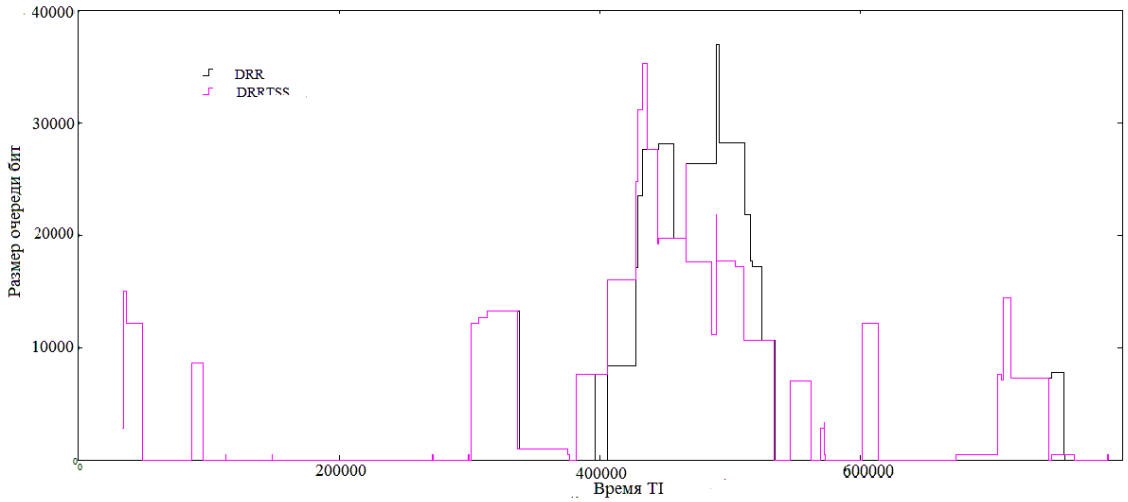


Рис. 2. Профиль изменения очередей

Результаты обработки статистической обработка данных об размерах очередей трафика различных классов QoS, полученных в результате моделирования DRR и DRR-TSS алгоритмов диспетчеризации очередей, приведены в табл. 2.

Таблица 2

Средние максимальные значения и очередей

Поток	Класс QoS	DRR		DRR–TSS	
		Максимальное значение размера очереди	Среднее значение размера очереди	Максимальное значение размера очереди	Среднее значение размера очереди
1	0–1	34136	3037	24352	2733
	2–3	70360	6865	58184	5828
	4–5	51336	8027	89576	8734
	6–7	71864	10425	71864	11047
	Общая максимальная очередь	71864		89576	
	Вариация средних очередей		7388		8314
	Средняя общая очередь		28354		28342
2	0–1	97848	10172	62264	7559
	2–3	71032	6998	71032	7486
	4–5	48192	5639	60648	6959
	6–7	59688	5548	68280	6366
	Общая максимальная очередь	97848		71032	
	Вариация средних очередей		10384		1193
	Средняя общая очередь		28357		28370

Анализ показывает, что при нагрузке, согласованной с выделенной полосой пропускания для трафика различных классов QoS алгоритм DRR-TSS приводит возрастанию вариации средних значений джиттера трафика различных классов на 12,5 % и увеличению

максимальной задержки на 25 %, но в случае рассогласования нагрузки алгоритм DRR-TSS уменьшает вариацию средних значений джиттера трафика различных классов на в 8,7 раз и снижает максимальную задержку на 28 %, при этом средний размер общей очереди практически не зависит от алгоритма диспетчеризации и вида трафика.

Таким образом, можно утверждать, что алгоритмы динамической диспетчеризации обеспечивают более комфортные условия передачи трафика в случае разбалансирования выделяемой для них полосы пропускания, чем классические циклические алгоритмы, что достигается за счет ограничения время непрерывной передачи данных при обслуживании каждой очереди и «перемешивание» кадров различных классов QoS трафика.

Библиографический список

1. Олифер, В. Компьютерные сети. Принципы, технологии, протоколы / В. Олифер, Н. Олифер. – СПб. : Питер, 2010. – 992 с.

2. Кучерявый, Е. Управление трафиком и качество обслуживания в сети Интернет / Е. А. Кучерявый. – СПб. : Наука и Техника, 2004. – 336 с.

3. Кизилов, Е. А. Моделирование адаптивной диспетчеризации очередей в коммутаторе с поддержкой QoS / Е. А. Кизилов, Н. Н. Коннов, Д. В. Патунин // Модели, системы, сети в экономике, технике, природе и обществе. – 2017. – № 2 (22). – С. 170–183.

4. Кизилов, Е. А. Эффективность «дефицитных» алгоритмов диспетчеризации очередей в коммутаторах с поддержкой QoS / Е. А. Кизилов, Н. Н. Коннов, Д. В. Патунин // Информатика: проблемы, методология, технологии : сб. материалов XVII Междунар. науч.-метод. конф. : в 5 т. – Воронеж : Вэлборн, 2017. – Т. 3. – С. 95–100.

5. Jensen, K. Coloured Petri Nets: modelling and validation of concurrent systems / K. Jensen, L. Kristensen. – Springer Science & Business Media, 2009. – P. 125.

6. Кизилов, Е. А. Моделирование коммутаторов Ethernet в промышленных сетях / Е. А. Кизилов, Н. Н. Коннов, В. Б. Механов // Новые информационные технологии и системы : тр. XI Междунар. науч.-техн. конф. – Пенза : Изд-во ПГУ, 2014. – С. 326–329.

7. Kizilov, E. Modeling of QoS in the industrial Ethernet switches / N. Konnov, D. Pashchenko, D. Trokoz // 2015 The 5th International Workshop on Computer Science and Engineering-Information Processing and Control Engineering (WCSE 2015-IPCE). – Moscow, Russia, 2015, April 15–17. – P. 185–190.

8. Никишин, К. Генератор трафика Ethernet на основе цветных сетей Петри / К. И. Никишин, Н. Н. Коннов // Модели, системы, сети в экономике, технике, природе и обществе. – 2016. – № 1 (17). – С. 299–307.

ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ И СЕТЕЙ

В. Ю. Егоров

ООО НТП «Криптософт», Пенза, Россия

ПЕРСПЕКТИВЫ ПРИМЕНЕНИЯ ОПЕРАЦИОННОЙ СИСТЕМЫ QP ОС В РАБОЧИХ СТАНЦИЯХ

Представлено описание операционной системы QP ОС для применения в качестве системы для рабочих станций. Дается общее представление о возможностях системы.

Научно-техническое предприятие «Криптософт» в течение более 20 лет разрабатывает и внедряет операционную систему QP ОС, а также вычислительные комплексы на ее основе.

В течение 2017 г. в составе QP ОС произошли существенные изменения, касающиеся, в первую очередь, внешнего вида системы. В системе появился полноценный рабочий стол, разработаны графические приложения. Теперь на рабочем столе QP ОС отображается список запущенных задач, можно размещать иконки для запуска приложений и формировать перечень необходимых программ. Также рабочий стол системы поддерживает экранные виджеты. Внешний вид системы представлен на рис. 1.

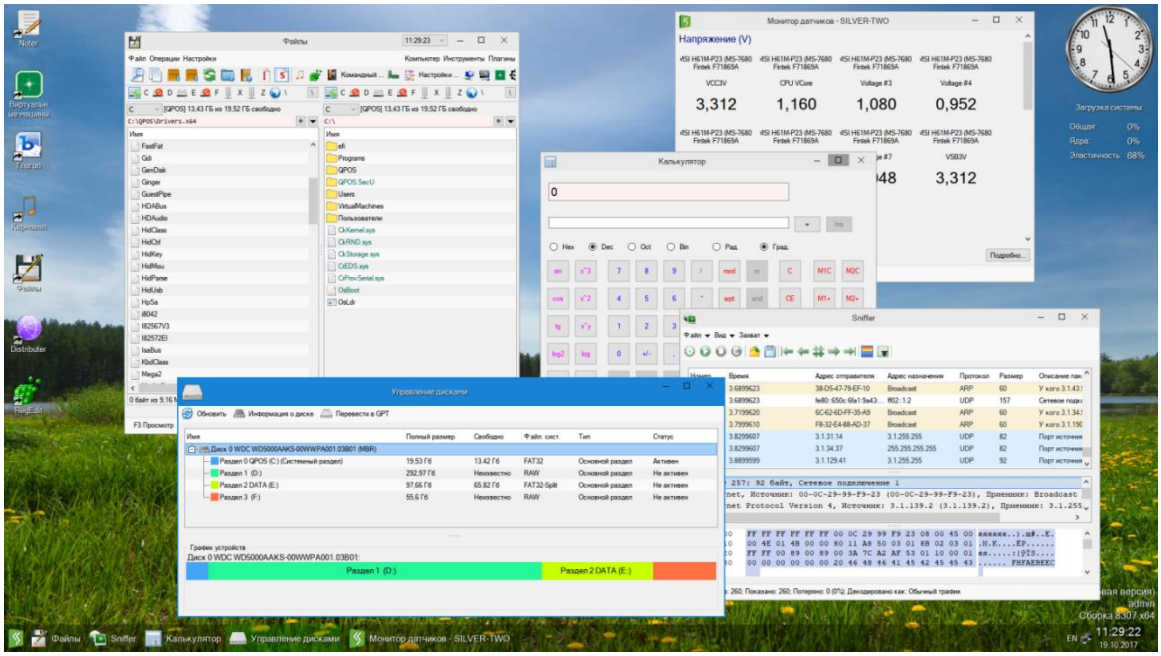


Рис. 1. Внешний вид операционной системы QP ОС

Как уже не раз отмечалось, QP ОС представляет собой универсальную операционную систему общего назначения, полностью разработанную на предприятии «Криптософт». QP ОС не является клоном какой бы то ни было другой операционной системы.

Все драйверы в составе QP ОС, как низкоуровневые, так и высокоуровневые, были написаны заново силами специалистов НТП «Криптософт». В состав драйверов также входят стеки сетевых протоколов TCP/IP 4 и 6 версий, а также протоколы SMB версий 1 и 2.

QP ОС сертифицирована ФСБ России для работы со сведениями, составляющими государственную тайну.

QP ОС обладает рядом преимуществ, позволяющих решать ответственные задачи по построению информационно-вычислительных систем. Среди них такие преимущества как:

- защищенная платформа;
- доверенная среда исполнения программ и операционных систем;
- быстродействие на уровне современных аналогов;
- комплексный подход к обеспечению безопасности.

QP ОС обладает развитой подсистемой безопасности. К компонентам безопасности в составе QP ОС относятся:

- дискреционный контроль доступа;
- мандатный контроль доступа;
- мандатное управление доверием;
- контроль целостности среды исполнения.

В настоящее время для операционной системы QP ОС разрабатывается набор офисных приложений. Одно из таких приложений – редактор документов Глагол. Программа Глагол, равно как и другие приложения в системе, поддерживает вывод документов на печать. Глагол будет активно развиваться в течение следующего года для обеспечения полноценной возможности создания документов. Внешний вид программы Глагол, а также поддержка вывода документов на печать показаны на рис. 2.

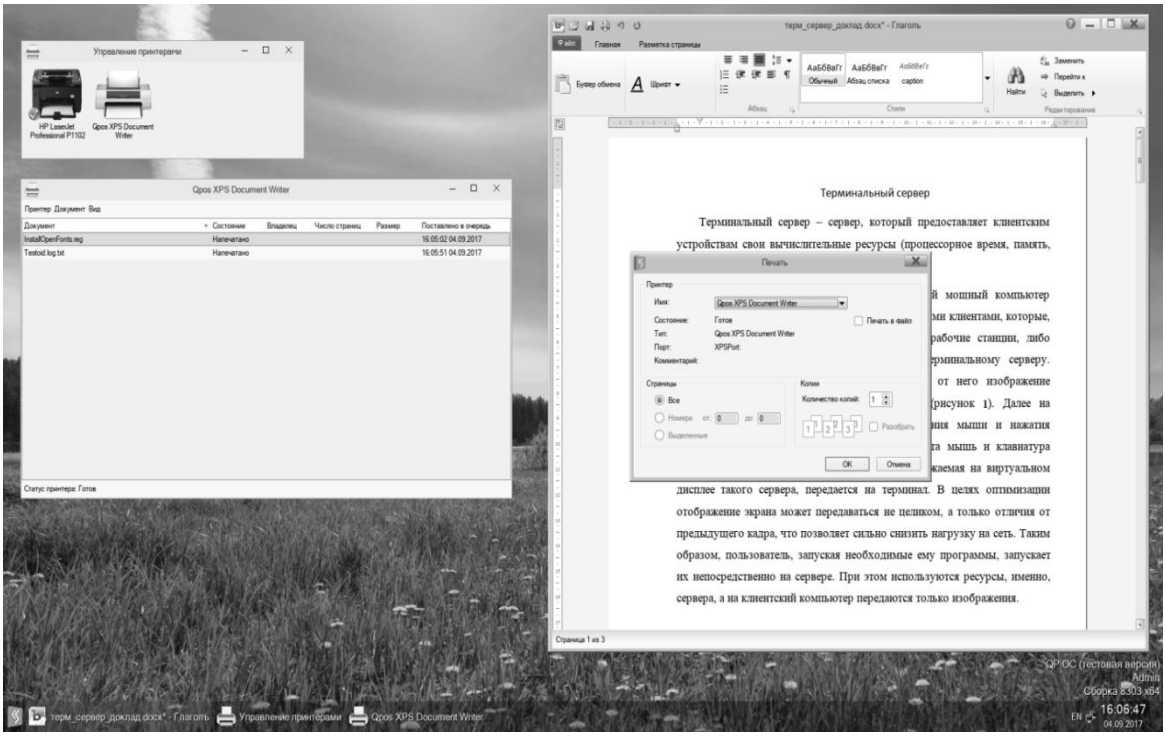


Рис. 2. Вывод документов на печать из программы Глагол

Поддержка мультимедиа в системе QP ОС реализована посредством программы Карнавал. Это медиапроигрыватель, который сейчас позволяет воспроизводить звуковые файлы различных форматов. Программа Карнавал представлена на рис. 3.

В составе операционной системы QP ОС появился браузер под названием Интернет. В настоящее время он обеспечивает базовый функционал по просмотру веб-страниц.

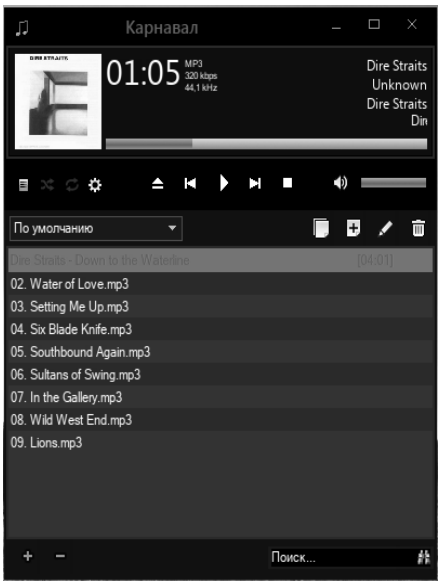


Рис. 3. Медиапроигрыватель Карнавал

Операционная система QP ОС постоянно и динамично развивается. Развитие происходит сразу по нескольким направлениям: это и дальнейшее усовершенствование ядра системы, и поддержка других аппаратных платформ, и появление новых прикладных программ. В ближайшее время по направлению применения QP ОС в качестве системы для прикладных программ планируется внедрение поддержки сканеров, просмотра видео, а также приложений для работы с презентациями и электронными таблицами.

Библиографический список

1. Официальная страница операционной системы QP ОС. – URL: http://cryptosoft.ru/qpos.html
2. Операционная система QP ОС на конференции OS DAY 2017. – URL: http://osday.ru/egorov.html#speaker#speaker

А. И. Иванов[1], А. В. Безяев[2], А. Г. Банных[3]

[1]Пензенский научно-исследовательский электротехнический институт, [2]Пензенский филиал ФГУП унитарного предприятия «Научно-технический центр «Атлас»,
[3]Пензенский государственный университет, Пенза, Россия

ХИ-КВАДРАТ ПРИБЛИЖЕНИЕ РАСПРЕДЕЛЕНИЯ ОШИБОК ВЫЧИСЛЕНИЯ ЭНТРОПИИ БИОМЕТРИЧЕСКОГО ОБРАЗА «ЧУЖОЙ» В ПРОСТРАНСТВЕ СВЕРТОК ХЭММИНГА

Целью работы является статистическое описание ошибок вычисления энтропии длинных кодов с зависимыми разрядами на малой тестовой выборке. Показано, что при переходе в пространство расстояний Хэмминга наблюдается логарифмическое сжатие размеров исходного алфавита спектра выходных состояний нейросетевой молекулы преобразователя биометрия-код. Это в конечном итоге позволяет выполнять быстрое тестирование преобразователей через вычисление их энтропии.

Обычно энтропию объекта исследования оценивают через наблюдение вероятности появления возможных состояний заранее заданного алфавита по Шеннону [1]. Когда число

состояний мало, проблем с оценкой энтропии не возникает. Этот классический подход вполне применим к оценке энтропии состояний преобразователей биометрия-код. Например, такие преобразователи за рубежом строят с использованием так называемых «нечетких экстракторов» [2–5]. Убедиться в этом можно, воспользовавшись достоверными биометрическими данными, самостоятельно получаемыми в среде моделирования «БиоНейроАвтограф» [6]. Этот продукт позволяет получать 416-ти мерные вектора биометрических параметров и соответствующие им 256 битные коды.

Проблема оценки энтропии «нечетких экстракторов» технически разрешима из-за того, что «нечеткие экстракторы» имеют короткий выходной код. Так, если в «нечетком экстракторе» при обработке 416 параметров применен код с 35 кратной избыточностью, то длина его информационной части составит не более 12 бит. Для 12 битных кодов проблем с оценкой их энтропии по Шеннону не возникает.

Однако, если воспользоваться российско-казахстанским опытом и применить нейросетевые преобразователи биометрия-код [7, 8], то длина выходного кода составит 256 бит и применить алгоритм Шеннона для вычисления энтропии станет невозможным.

Выход из этого положения состоит в переходе из пространства вероятности появления обычных кодов в пространство появления расстояний Хэмминга [9, 10]. На рис. 1 дана схема вычислений, рекомендуемая ГОСТ Р 52633.3.

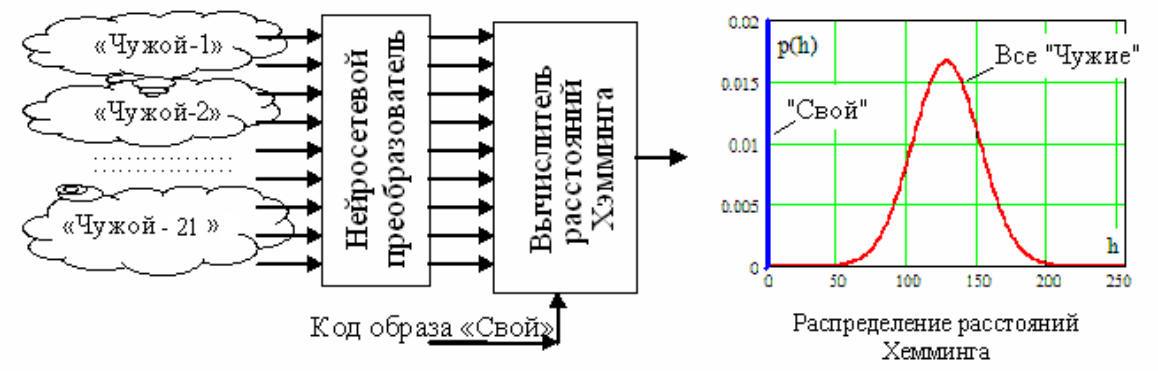


Рис. 1. Вариант схемы и результаты тестирования преобразователя биометрия-код при переходе в пространство расстояний Хэмминга

Нейронная сеть среды моделирования «БиоНейроАвтограф» [6] преобразует 416 биометрических параметров в код длиной 256 бит. То есть его выходные коды могут давать 2^{256} состояний, что не позволяет оценить вероятности их появления. Однако, если перейти в пространство расстояний Хэмминга между кодами образов «Чужой-1», «Чужой-2», …., «Чужой-21» и кодом образа «Свой», то мы получим спектр, имеющий всего 256 возможных состояний. Наблюдается логарифмическое свертывание числа возможных спектральных состояний $\log_2(2^{256}) = 256$ и, следовательно, экспоненциальное упрощение задачи оценки энтропии.

Все криптографические операции создаются таким образом, чтобы давать данные близкие к белому шуму. То есть разряды кодов независимые, а энтропия кодов точно совпадает с длиной кода. Для биометрических данных все иначе, их разряды зависимы (коррелированны). Теоретически 256 зависимых бит можно заменить на меньшее число независимых бит, например путем применения процедуры декорреляции.

Стандартная процедура вычислений по ГОСТ Р 52633.3 построена на том, что подав на нейросеть 21 образ «Чужой», получается 21 код «$\bar{x}$». Образ «Свой» дает код «$\bar{c}$», соответственно можно вычислить расстояние Хэмминга сложением по модулю два их разрядов:

$$h = 256 - \sum_{i=0}^{255} ("c_i") \oplus ("x_i") \qquad (1)$$

Далее по выборке в 21 опыт мы можем вычислить математическое ожидание – $E(h)$ и стандартное отклонение – $\sigma(h)$. Знание о двух статистических моментах позволяет оценить вероятность угадывания кода «Свой» – P_2, подставляя случайные образы «Чужой»:

$$P_2 \approx \frac{1}{\sigma(h)\sqrt{2\pi}} \int_0^1 \exp\left\{\frac{-\left(E(h)-u\right)^2}{2\left(\sigma(h)\right)^2}\right\} \cdot du. \quad (2)$$

В этом случае энтропия нейросетевого преобразователя оценивается следующим образом:

$$H("x_1, x_2, ..., x_{256}") \approx -\log_2(P_2). \quad (3)$$

Каждая достаточно сложная вычислительная процедура, как правило, накапливает ошибки исходных данных. Для оценки показателя обусловленности последовательности процедур (2), (3) необходимо использовать большую базу образов в 300 выборок по 21 образу «Чужой». В результате тестирования на таком объеме данных математическое ожидание составляет – E(h) = 107,26 бит, а его стандартное отклонение – σ{E(h)} = 6,77 бит. Это означает, что на выборке в 21 опыт относительная ошибка вычисления математического ожидания может составлять до 18,9 % при нормальном законе распределения значений.

Математическое ожидание стандартного отклонения составляет E{σ(h)} = 27,84 бит, а его стандартное отклонение σ{σ(h)} = 4,04 бит. Это означает, что стандартное отклонение на выборке в 21 опыт может оцениваться с относительной ошибкой до 43,5 % при нормальном законе распределения значений. Все это приводит к итоговым относительным ошибкам вычисления энтропии до 271 %. Мы имеем дело с плохо обусловленной задачей.

На рис. 2 приведен пример гистограммы распределения значений энтропии, вычисленной по 300 выборкам, каждая из которых состоит из 21 опыта.

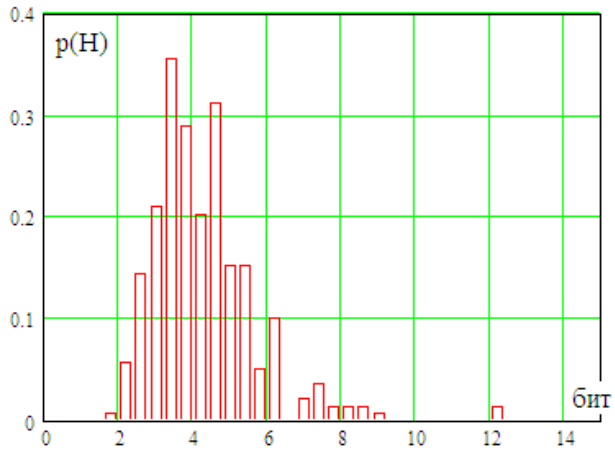


Рис. 2. Гистограмма распределения значений энтропии, вычисленной на выборке в 21 опыт

Из рис. 2 видно, что при математическом ожидании энтропии – E(H)=4.52 бита (данные численного эксперимента) минимальное значение энтропии составляет 1,8 бита, а максимальное значение энтропии составляет 12,2 бита. Максимальная оценка оказывается выше математического ожидания на 271 %, что еще раз подтверждает плохую обусловленность решаемой задачи. Относительная ошибка вычислений нелинейно усиливается и оказывается много больше, чем сумма средних значений максимумов относительных ошибок исходных данных:

$$cond(H) \approx \frac{2\max(\left|E(H)-H_i\right|)}{(\max(\left|E(h)-h_i\right|)+\max(\left|E(\sigma(h))-\sigma(h_i)\right|)} \approx 8{,}96. \quad (4)$$

Наблюдается почти 9-ти кратное усиление ошибок исходных данных при стандартных вычислениях по ГОСТ Р 52633.3 [10] на малой выборке в 21 опыт. Для снижения столь

существенного эффекта накопления ошибок на порядок потребуется увеличить объем тестовой выборки на 2 порядка до 2100 опытов.

В силу того, что число обусловленности вычисления энтропии в пространстве сверток Хэмминга оказывается высоко (4) необходимо проводить регуляризацию подобных вычислений [11–13]. Для этой цели необходима априорная информация о законе распределения ошибок вычисления энтропии по формуле (3).

Проведенный численный эксперимент для 1000 выборках по 8 опытов, показал, что закон распределения плотности вероятности значений близок к хи-квадрат распределению. Соответствующие графики распределений приведены на рис. 3.

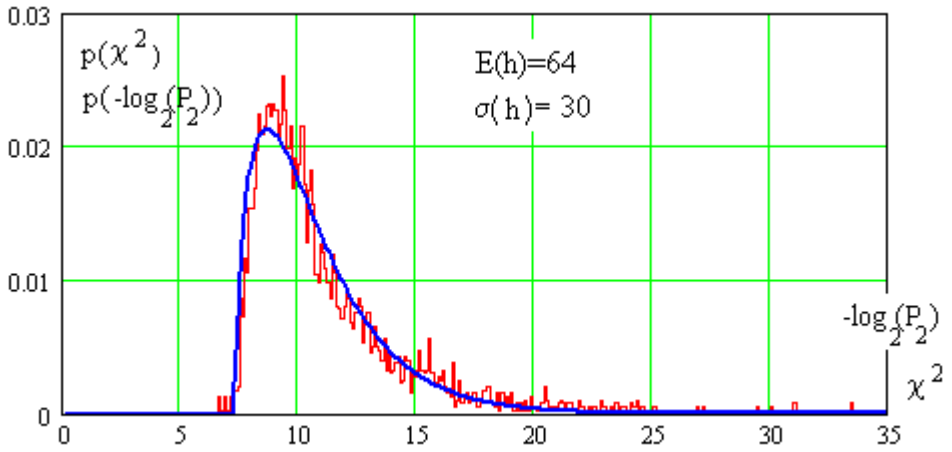


Рис. 3. Распределение вероятности ошибок вычисления энтропии на малой выборке из 8 опытов для нормального закона распределения значений расстояний Хэмминга с математическим ожиданием $E(h) = 64$ и стандартным отклонением $\sigma(h) = 30$

Из рис. 3 видно, что реальное распределение ошибок (кривая с шумом) хорошо описывается смещенным на величину 7,278 бит хи-квадрат распределением с числом степеней свободы $m = 3{,}6$ (сплошная линия).

Таким образом, можно сделать вывод, что плотность распределения ошибок вычисления энтропии на малых выборках хорошо описывается хи-квадрат распределениями с дробным (фрактальным) числом степеней свободы. Пользуясь этим, можно дать достаточно точное аналитическое описание распределения ошибок вычислений, которое позднее может быть использовать на следующих этапах регуляризации.

Библиографический список

1. Яглом, А. М. Вероятность и информация / А. М. Яглом, И. М. Яглом. – М. : ДомКниги, 2007. − 512 с.

2. Juels, A. Fuzzy Commitment Scheme / A. Juels, M. A. Wattenberg // Proc. ACM Conf. Computer and Communications Security. – 1999. − P. 28–36.

3. Monrose, F. Cryptographic key generation from voice / F. Monrose, M. Reiter, Q. Li, S. Wetzel // In Proc. IEEE Symp. on Security and Privacy. – 2001.

4. Dodis, Y. Fuzzy Extractors: How to Generate Strong Keys from Biometrics and Other Noisy / Y. Dodis, L. Reyzin, A. Smith // In EUROCRYPT. – 2004. − P. 523–540.

5. Hao, F. Crypto with Biometrics Effectively / Feng Hao, Ross Anderson, John Daugman // IEEE TRANSACTIONS ON COMPUTERS. – 2006. – Vol. 55, № 9, September

6. Иванов, А. И. Среда моделирования «БиоНейроАвтограф» / А. И. Иванов, О. С. Захаров. – 2009. – URL: http://пниэи.рф/activity/science/noc.htm

7. Волчихин, В. И. Быстрые алгоритмы обучения нейросетевых механизмов биометрико-криптографической защиты информации : моногр. / В. И. Волчихин, А. И. Иванов, В. А. Фунтиков. – Пенза : Изд-во ПГУ, 2005. − 273 с.

8. Технология использования больших нейронных сетей для преобразования нечетких биометрических данных в код ключа доступа : моногр. / Б. С. Ахметов, А. И. Иванов, В. А. Фунтиков, А. В. Безяев, Е. А. Малыгина. – Казахстан, Алматы : ТОО «Изд-во LEM», 2014. − 144 с. – URL: http://portal.kazntu.kz/files/publicate/2014-06-27-11940.pdf

9. Малыгин, А. Ю. Быстрые алгоритмы тестирования нейросетевых механизмов биометрико-криптографической защиты информации : моногр. / А. Ю. Малыгин, В. И. Волчихин, А. И. Иванов, В. А. Фунтиков. – Пенза : ПГУ, 2006. − 161 с.

10. ГОСТ Р 52633.3–2011 Защита информации. Техника защиты информации. Тестирование стойкости средств высоконадежной биометрической защиты к атакам подбора.

11. Тихонов, А. Н. Методы решения некорректных задач / А. Н. Тихонов, В. Я. Арсенин. – М. : Наук, 1984. − 256 с.

12. Бабушинский, А. Б. Некорректные задачи. Численные методы и приложения / А. Б. Бабушинский, А. В. Гончарский. – М. : Изд-во Моск. ун-та, 1989. − 199 с.

13. Васин, В. В. Некорректные задачи с априорной информацией / В. В. Васин, А. Л. Агеев. – Екатеринбург : УИФ «Наука», 1993.− 264 с.

В. И. Волчихин[1], А. И. Иванов[2], С. Е. Вятчанин[1]

[1]Пензенский государственный университет,
[2]Пензенский научно-исследовательский электротехнический институт, Пенза, Россия

ОСОБЕННОСТИ ОБУЧЕНИЯ СЕТЕЙ ВЕРОЯТНОСТНЫХ НЕЙРОНОВ «КРАМЕРА – ФОН МИЗЕСА» НА МАЛЫХ БИОМЕТРИЧЕСКИХ ВЫБОРКАХ

Синтезированный искусственный нейрон «Крамера – фон Мизеса» унаследовал от квадратичных форм устойчивость алгоритма обучения. Сети нейронов «Крамера – фон Мизеса» дают выходной код для образов «Чужой» с низкой энтропией из-за отсутствия баланса состояний «0» и «1» в его разрядах. Этот недостаток предложено устранить путем применения выходного квантователя на выходе сумматоров нейронов «Крамера-фон Мизеса» с тремя выходными состояниями. Это позволяет рассматривать данные сети нейронов как перспективу для следующего поколения нейросетевых преобразователей биометрия-код.

Существует множество статистических критериев для поверки гипотезы нормального распределения значений биометрических данных. Наиболее часто на практике используется хи-квадрат критерий [1] в силу того, что для этого критерия Пирсон в 1900 г. построил аналитическое описание хи-квадрат распределения. К сожалению, хи-квадрат критерий оказывается работоспособен только на выборках в 200 опытов и больше. При обработке биометрических данных приходится иметь выборки в 20 примеров, что мешает применять хи-квадрат критерий.

Такая же ситуация возникает и случаи использования других не параметрических статистических критериев [2], все они так же дают надежные оценки при выборках существенно более 20 примеров.

Для биометрии данные одного контролируемого параметра примеров образа «Свой» имеют распределение близкое к нормальному. Для данных случайно выбранных образов «Чужие» распределение оказывается близко к равномерному. В связи с этим мы можем исследуемые критерии оценивать по отношению к друг другу сравнивая между собой, даваемые ими равные ошибки первого и второго рода $P_1 = P_2 = P_{EE}$ при использовании малых выборок нормального и равномерного законов распределения. На рис. 1 приведены функции снижения вероятностей ошибок по мере роста размеров тестовой выборки. Из рис. 1 видно, что мощность критерия KfM при малых выборках оказывается выше, чем у хи-квадрат критерия. Это делает критерий KfM более перспективным для биометрии в сравнении с хи-квадрат критерием, критерием Фроцини и критерием Джини.

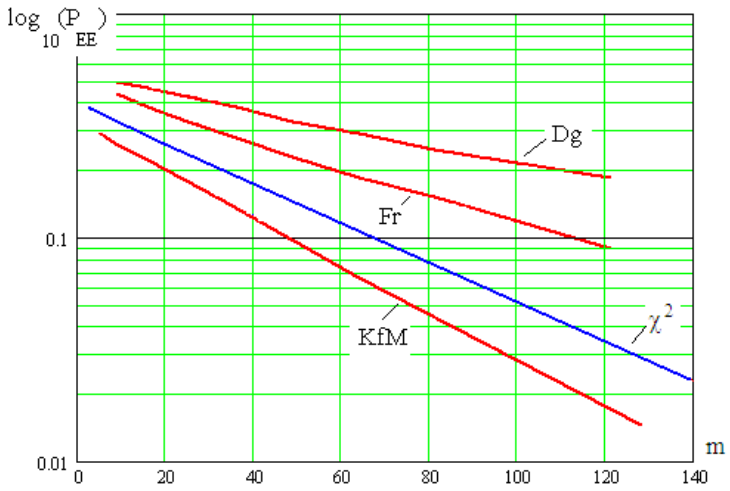


Рис. 1. Сопоставление мощностей интегральных статистических критериев Джини (Dg), Фроцини (Fr), хи-квадрат (χ2) и Крамера – фон Мизеса (KfM)

Видимо основной причиной высокой мощности «Крамера – фон Мизеса» (KfM) критерия является то, что у него оказываются минимальные шумы квантования континуальных данных.

Исходный критерий «Крамера-фон Мизеса» вычисляется через интеграл квадрата разницы между гипотетической функцией вероятности – $P(\nu)$ эмпирической функции вероятности $\hat{P}(\nu)$:

$$KfM = \int_{-\infty}^{+\infty} (P(v) - \hat{P}(v))^2 dv. \tag{1}$$

Формальный переход от интеграла (1) к операции суммирования нейроном с n входами не сложен:

$$\begin{cases} KfM(\nu) = \sum_{i=1}^{n} (P(v_i) - \hat{P}(v_i))^2 \\ KfM(\xi) = \sum_{i=1}^{n} (P(v_i) - \hat{P}(\xi_i))^2 \end{cases}, \tag{2}$$

где ν_i пример образа «Свой» по n-контролируемым нейроном биометрическим параметрам; ξ_i пример образа «Чужой» по n-контролируемым нейроном биометрическим параметрам.

Эталонная функция вероятности – $P(\nu)$ оказывается легко вычислима только в том случае, когда выполнено центрирование и нормирование, контролируемых биометрических параметров образа «Свой»:

$$\tilde{\nu}_i = \frac{\nu_i - E(\nu_i)}{\sigma(\nu_i)}. \tag{3}$$

В новой центрированной и нормированной системе биометрических координат эталонная функция вероятности имеет нулевое математическое ожидание и единичное стандартное отклонение:

$$P_\nu(\tilde{v}) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\tilde{v}} \exp\left\{\frac{-u^2}{2}\right\} du. \tag{4}$$

На рис. 3 дан пример эталонной функции вероятности для нормального закона.

Из рис. 3 видно, что данные примеров образа «Свой» после их преобразования в вероятность должны оказываться в интервале от 0.0 до 1.0. Причем появление предельных значений 0.0 и 1.0 маловероятно.

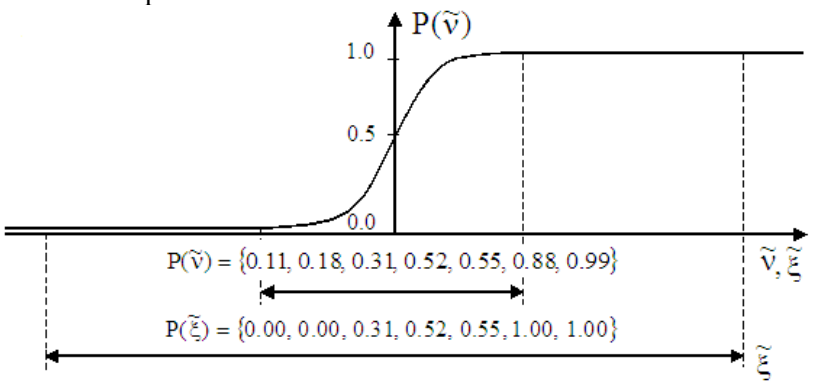


Рис. 3. Работа функции преобразования вектора значений переменных в вектор значений их вероятности

Совершенно иная ситуация возникает, если вероятностному нейрону «Крамера – фон Мизеса» будут предъявлены данные примера образа «Чужой». Как видно из рис. 3 динамический диапазон данных – $\tilde{\xi}_i$ примерно в три раза больше, чем данных образа «Свой». Это означает, что на входы нейрона KfM будут поступать несколько предельных минимальных значений вероятности {0.0, 0.0,….} в начале упорядоченного списка и несколько предельных значений в конце упорядоченного списка {….., 1.0, 1,0}.

В целом вероятностный нейрон KfM работает за счет того, что выявляет неоправданно большое число предельных состояний вероятности в наблюдаемом биометрическом образе. Нейрон KfM и структура нейронной сети, сформированной из нейронов KfM представлены на рис. 4.

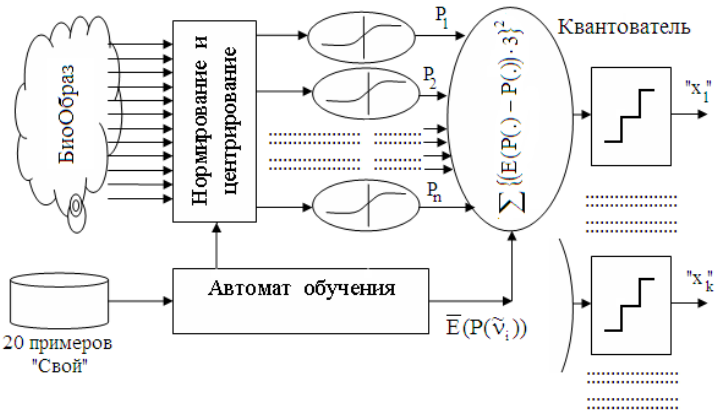


Рис. 4. Обобщенная структура нейросетевого преобразователя биометрических данных в код, построенная на использовании вероятностных нейронов «Крамера – фон Мизеса»

При использовании трехуровнего квантования [3, 4] (см. рис. 4) вероятности состояний «0» и «1» на выходе разрядов сети нейронов KfM выравниваются. Наблюдается рост энтропии выходных кодов «Чужой» до величины порядка 45 бит, что в полтора раза больше, чем тот же показатель для линейных нейронов, обученных по ГОСТ Р 52633.5 [5].

Библиографический список

1. Р 50.1.037–2002 Рекомендации по стандартизации. Прикладная статистика. Правила проверки согласия опытного распределения с теоретическим. Ч. I. Критерии типа χ2. – М., 2001. − 140 с.

2. Р 50.1.037–2002 Прикладная статистика. Правила проверки согласия опытного распределения с теоретическим. Ч. II. Непараметрические критерии. – М., 2002. − 123 с.

3. ГОСТ Р 52633.5–2011 Защита информации. Техника защиты информации. Автоматическое обучение нейросетевых преобразователей биометрия-код доступа. – М., 2011.

А. И. Иванов[1], Е. А. Малыгина[2]

[1]Пензенский научно-исследовательский электротехнический институт,
[2]Пензенский государственный университет, Пенза, Россия

ОСОБЕННОСТИ МНОГОМЕРНОЙ СТАТИСТИЧЕСКОЙ ОБРАБОТКИ БИОМЕТРИЧЕСКИХ ДАННЫХ

Использование многомерной статистической обработки малых выборок биометрических данных позволяет получать решения с достаточно высокой достоверностью.

Популярность использования хи-квадрат критерия Пирсона [1] во многом обусловлена тем, что при $n\to\infty$ его распределение описывается через гамма функцию с $m = k–1$ числом степеней свободы:

$$p_{\chi^2}(n=\infty, m=k-1, x)=\frac{1}{2^{\frac{m}{2}}\cdot\Gamma\left(\frac{m}{2}\right)}\cdot x^{\frac{m}{2}-1}\cdot e^{\frac{-x}{2}}.$$

Применение хи-квадрат критерия для многомерных и сильно коррелированных биометрических данных дает неудовлетворительные результаты [2].

Так для принятия решений с уровнем доверия 0,99 придется использовать выборку, состоящую из 400 результатов испытаний. Применение столь больших тестовых выборок недопустимо для биометрии, поэтому необходимо добиться их снижения примерно на порядок.

Известно, что биометрические данные многомерны [3]. Например, нейросетевой преобразователь биометрия-код [3], преобразует 416 рукописных биометрических параметров в код личного ключа длиной 256 бит. То есть имеется возможность анализировать не один, а 416 биометрических параметров написания слова-пароля. При этом можно увеличить объем обрабатываемых данных и повысить достоверность принимаемых решений.

Для многомерной обработки воспользуемся сложением частных критериев Пирсона:

$$\chi^2(\nu_1,\nu_2,...,\nu_q)=\frac{\chi^2(\nu_1)+\chi^2(\nu_2)+...+\chi^2(\nu_q))}{q}. \quad (1)$$

Преобразование (1) эквивалентно использованию сети частных критериев Пирсона. Сеть частных хи-квадрат критериев Пирсона имеет входные и выходные нелинейные преобразования при линейном суммировании данных между ними.

При переходе к моделированию сети частных критериев хи-квадрат Пирсона достаточно вместо двух программных генераторов псевдослучайных чисел использовать q пар генераторов. При малой размерности входных данных $q \le 16$ особых сложностей в программировании численного эксперимента не возникает.

При многомерной статистической обработке биометрических данных отпадает необходимость вычисления многомерной корреляционной матрицы.

С ростом размерности решаемой задачи все более актуальным становится процедура ее симметризации. В нейросетевой биометрии, принято [4] осуществлять симметриза-

цию влияния корреляционных связей через вычисление математического ожидания модулей парных коэффициентов корреляции. Для этого берется несколько сотен пар случайно выбранных биометрических параметров. При любой размерности задачи, удается заменить реальные данные их некоторым эквивалентом, имеющим одинаковые корреляционные связи между всеми заданными параметрами. Такое упрощение задачи позволяет легко оценить эквивалентный показатель размерности:

$$\tilde{q} \approx (q-1)\cdot(1-R^2)+1. \qquad (2)$$

Чем выше корреляция данных, тем меньше оказывается эквивалентная размерность, обработки статистических данных.

Рост размерности обработки и рост коррелированности данных работают в противоположных направлениях. Всегда можно понизить размерность задачи, компенсируя тем самым влияние коррелированности данных. То есть, пользуясь преобразованиями вида (2) вполне возможно свести задачу применения многомерных сетей Пирсона к зависимым данным к более простой задаче обработки независимых данных.

Соответствующие таблицы пересчета уже созданы для искусственных нейронных сетей и опубликованы в работе [4]. Как следствие, аналогичные таблицы преобразований могут быть построены и для сетей частных критериев Пирсона.

Библиографический список

1. Р 50.1.037–2002 Рекомендации по стандартизации. Прикладная статистика. Правила проверки согласия опытного распределения с теоретическим. Ч. I. Критерии типа χ2. – М., 2001. − 140 с.

2. Ахметов, Б. С. Оценка рисков высоконадежной биометрии : моногр. / Б. С. Ахметов, Д. Н. Надеев, В. А. Фунтиков, А. И. Иванов, А. Ю. Малыгин. – Алматы : КазНТУ, 2014. − 108 с.

3. Иванов, А. И. Многомерный статистический анализ качества биометрических данных на предельно малых выборках с использованием критериев среднего геометрического, вычисленного для анализируемых функций вероятности / А. И. Иванов, К. А. Перфилов, А. Ю. Малыгин // Измерения. Мониторинг. Управление. Контроль. – 2016. – № 1. – С. 64–72.

4. «БиоНейроАвтограф» – среда моделирования больших искусственных нейронных сетей, преобразующих данные рукописных образов в код пароля. – URL: http://пниэи.рф/activity/science/noc.htm

5. Ахметов, Б. С. Алгоритмы тестирования биометрико-нейросетевых механизмов защиты информации / Б. С. Ахметов, В. И. Волчихин, А. И. Иванов, А. Ю. Малыгин. – Казахстан, Алматы : КазНТУ им. Сатпаева, 2013. − 152 с. – URL: http://portal.kazntu.kz/files/publicate/2014-01-04-11940.pdf

К. М. Фролов[1], А. В. Дубравин[2]

[1]Пензенский государственный университет, Пенза, Россия
[2]ООО НТП «Криптософт», Пенза, Россия

ГОРИЗОНТАЛЬНОЕ МАСШТАБИРОВАНИЕ ДАННЫХ В СОВРЕМЕННЫХ СУБД

Приведены возможности распределенных систем, проблемы, возникающие при их внедрении, методы их разрешения. Также приведены данные о возможностях реализации горизонтального масштабирования на базе разрабатываемой в ООО НТП «Криптософт» системы управления базами данных.

Введение

С ростом любой компании, эксплуатирующей сервера баз данных, объем хранимой информации и нагрузка на систему увеличивается. При достижении определенной нагрузки, сервер базы данных перестает справляться с поставленным задачами. Одним из путей

решения является оптимизация архитектуры самой базы, внедрение дополнительных индексов или увеличение мощности ЭВМ, эксплуатируемой в качестве сервера. Однако все эти методы не помогут, если масштабы роста нагрузки на систему будут чересчур высокими. В этом случае на помощь может прийти горизонтальное масштабирование базы данных, то есть увеличение числа ЭВМ, решающих поставленную задачу. Выгода, достигаемая за счет использования такого рода архитектуры, очевидна – возможность повышения производительности системы путем добавления новой узловой ЭВМ и синхронизации данных с другими узлами. Пример распределенной системы представлен на рис. 1.

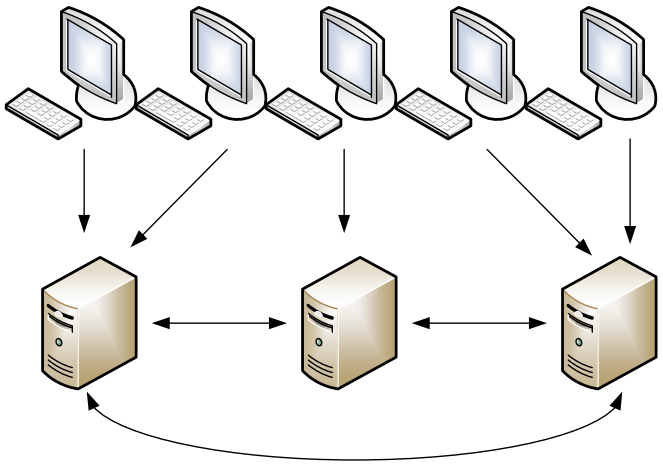


Рис. 1. Пример распределенной системы

Однако и недостатков такая система не лишена. Один из этих недостатков заключается в невозможности обойти CAP теорему, которая оперирует тремя свойствами распределенной системы:

- consistency (согласованность) предполагает, что, если данные успешно записаны в распределенное хранилище, то любой клиент при запросе получит именно актуальные данные;
- availability (доступность) указывает, что клиент в любой момент может получить данные из распределенного хранилища или сообщение об отсутствии данных, если они не были сохранены;
- partition tolerance (устойчивость к разделению системы) предполагает, что при потери сообщений между компонентами или выходе компоненты системы из строя сама система не теряет работоспособность.

В CAP теореме говорится, что при построении распределенных систем, можно удовлетворить только 2 из 3 вышеперечисленных свойств системы.

В реальных системах даже достижение 2 критериев из 3 весьма затруднительно, поскольку при разработке системы сказывается также и человеческий фактор. Сама CAP теорема лишь частично применима к современным распределенным системам [1]. В реальных системах ни один разработчик не станет жертвовать свойством устойчивости к разделению системы, поскольку в противном случае смысл распределенной архитектуры теряется. Таким образом при разработке распределенных систем пожертвовать можно только доступностью или согласованностью и реализовать оставшуюся часть.

Примером распределенных систем, в которых сочетаются свойства доступности к данным и устойчивости системы (AP-системы), является множество NoSQL решений. Считается, что в таких базах данных целостность данных не гарантирована. В случае таких систем обеспечивается так называемый целесообразный уровень целостности данных. Распределенные базы данных, построенные с помощью NoSQL решений, основываются на подходе, при котором требования к доступности и целостности данных выполняются не в полной мере. Такой подход получил акроним BASE (Basically Available, Soft-state, Eventually consistent) – базовая доступность, неустойчивое состояние, согласованность в конечном

счете. Этот подход является прямой противоположностью принципу ACID, который используется практически во всех современных реляционных базах данных.

Возможности реализации распределенной архитектуры

Для понимания возможностей реализации распределенной системы на базе разрабатываемой в ООО НТП «Криптософт» СУБД на рис. 2 приведена архитектура этой СУБД.

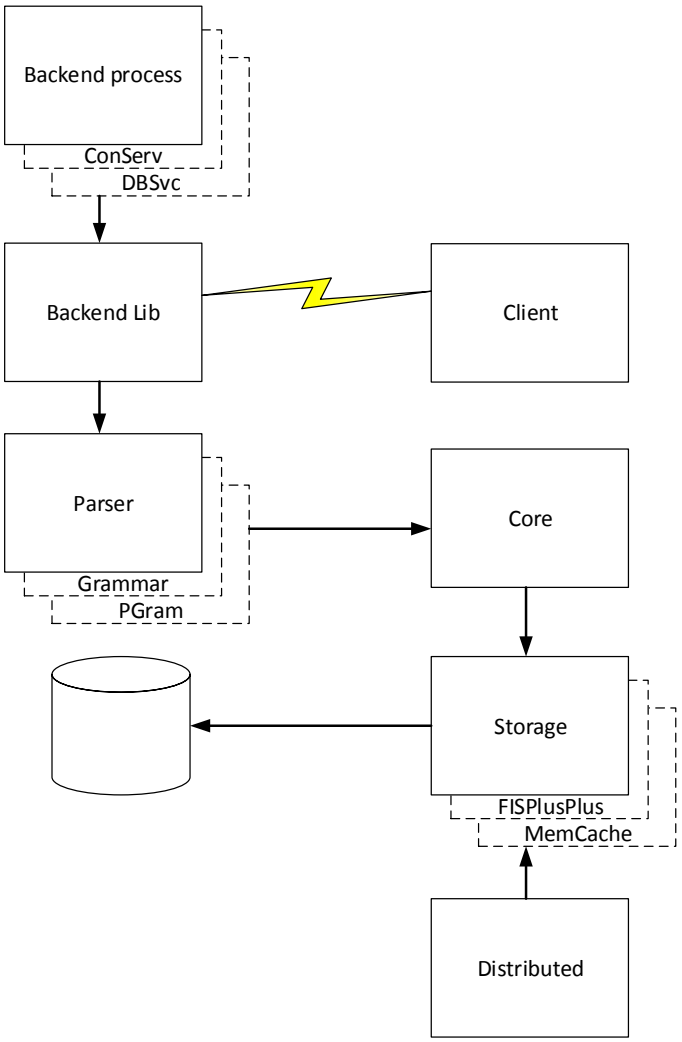


Рис. 2. Архитектура разрабатываемой СУБД

Так как в данной СУБД поддерживается работа с различными хранилищами, то добавление новых компонентов на этом уровне в архитектуру происходит безболезненно за счет использования интерфейсов. Для реализации возможности создания БД с распределенной архитектуры планируется провести декомпозицию задачи поддержания согласованного состояния данных в БД и решать отдельные подзадачи с помощью выделенных модулей.

Для решения проблемы отказоустойчивости и поддержания согласованности данных будет необходимо реализовать модуль поиска консенсуса, который должен будет поддерживать согласованное состояния данных во всей системе. Для поддержания согласованности данных на физическом уровне в разрабатываемой СУБД также планируется ввести новый тип хранилища, которое будет являться распределенным и защищенным. Кроме того, на каждом сервере БД в системе должны быть запущены собственные бэкенд-процессы (ConServ или DBSvc), поэтому потребуется внести в соответствующие модули

коррективы, учитывающие возможность использования соответствующих приложений в распределенной архитектуре.

Задача достижения консенсуса

Для понимания устройства распределенной базы данных на рис. 3 приведена схема типовой распределенной БД, являющейся AP-системой.

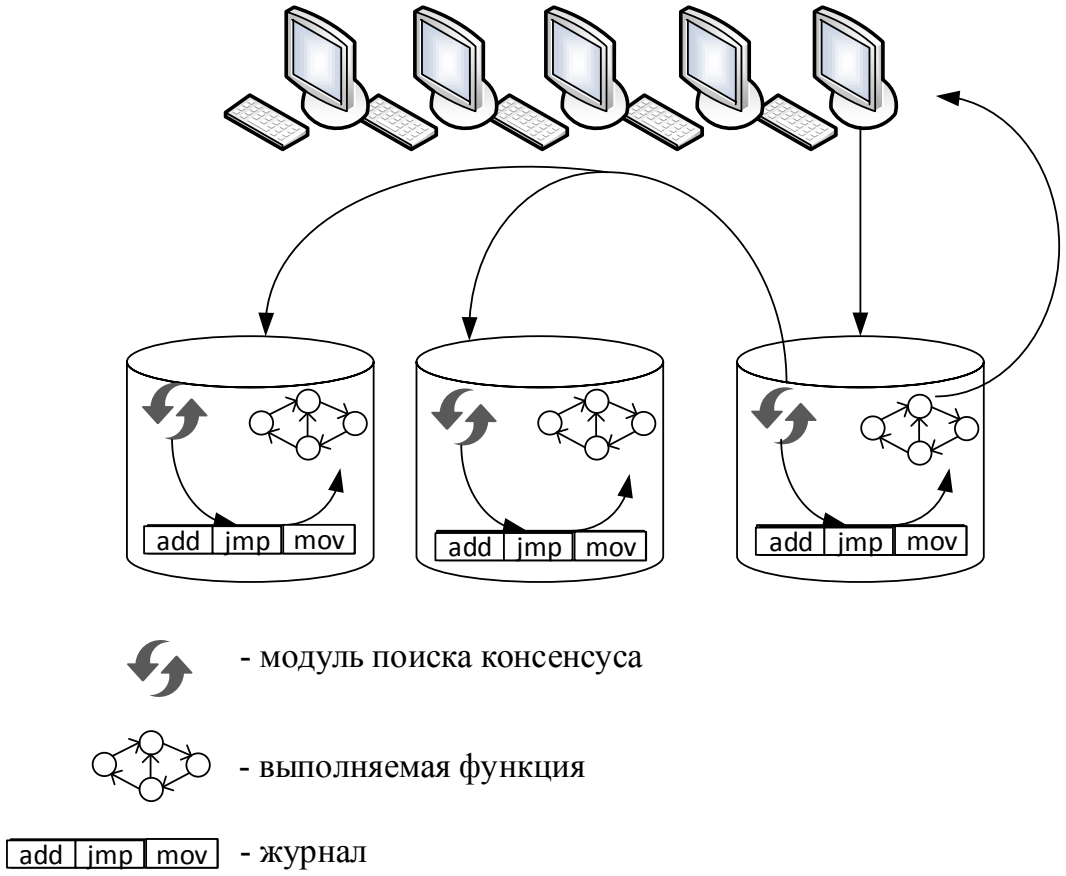


Рис. 3. Пример распределенной базы данных

Чтобы достигнуть свойства partition tolerance, при организации распределенной системы нужно принимать во внимание фактор отказов оборудования. Представим себе распределенную систему, состоящую из нескольких ×86 серверов. Для одного сервера вероятность отказа довольно мала, но в случае распределенной системы мы будем иметь дело с множеством серверов, а, следовательно, вероятность отказа растет в несколько раз [2].

Помимо вероятности отказа самих серверов также нужно закладывать вероятность отказа отдельных компонент, которые могут привести к несогласованности данных. На основе статистики, собранной в Google, утверждалось, что для кластера из 1800 машин имеет место быть порядка 1000 отказов в первый год эксплуатации, т.е. около 3 отказов в день. Таким образом, если отказоустойчивость не закладывается в ПО распределенной системы, то после разработки будет иметься система, в которой каждая из описанных причин может привести к отказу всей системы.

Задача достижения консенсуса заключается в получении согласованного значения группой участников распределенной системы, когда возможны отказы отдельных ее участков, предоставление этими участками некорректной информации или искажение информации самой средой передачи данных. Решается задача достижения консенсуса с помощью алгоритмов поиска консенсуса, наиболее популярными из которых являются PAXOS и RAFT.

Заключение

Горизонтального масштабирования данных в современных реалиях является одним из ключевых компонентов любой конкурентоспособной СУБД. Таким образом, реализация алгоритмов и компонентов, обеспечивающих горизонтальное масштабирование в СУБД, является актуальной задачей.

Библиографический список

1. Ефимов, С. Оптимизация структуры распределенных вычислительных систем / С. Ефимов. – М. : Lambert Academic Publishing, 2012. – 108 с.
2. Таненбаум, Э. Распределенные системы. Принципы и парадигмы / Э. Таненбаум, М. Ван Стеен. – СПб. : Питер, 2003. – 880 с.

В. В. Кормишина

Пензенский государственный университет, Пенза, Россия

ПРИМЕНЕНИЕ ЭЛЛИПТИЧЕСКИХ КРИВЫХ В КРИПТОГРАФИИ

Объектом исследования является возможность применения теории эллиптических кривых при реализации криптографических протоколов (сеансов) с использованием слабозащищенных, потенциально доступных для взлома каналов связи. В этом случае возникает необходимость дополнительной защиты передаваемой информации, особенно, если эта информация является изображением, и именно эллиптическая криптография создает возможность ее обеспечения. Были решены модельные задачи для различных изображений различной природы и цветовой насыщенности

В конце прошлого века на стыке теории алгоритмов, чисел и компьютерной алгебры зародилось и в настоящее время бурно развивается направление, известное нам как криптография с открытым ключом. Данный раздел криптографии позволяет успешно решать различные задачи, связанные с защитой информации, в компьютерных сетях.

В последние годы активное развитие получила так называемая криптография эллиптических кривых, которая была открыта независимо Нилом Коблицем и Виктором Миллером в 1985 году (также используется термин «эллиптическая криптография»). Роль основной криптографической функции (операции) здесь выполняет скалярное умножение точки эллиптической кривой на константу, которая, в свою очередь, определяется через операции сложения и удвоения точек эллиптической кривой. Последние выполняются на основе операций умножения, сложения и инвертирования в конечном поле, над которыми рассматривается кривая.

Криптография в современном мире имеет огромное значение, в частности от того, что ресурсы становятся доступными и многие из них требуют более сильной защиты. Постоянно меняются методы криптографии, однако и виды криптографических атак становятся разнообразными и позволяют за короткое время взломать практически любые данные. Мы должны разрабатывать более устойчивые методы для защиты данных, чтобы сохранить информацию и безопасно передать ее.

Эллиптическая криптография – принципиально новое направление, и ее применение в беспроводных коммуникациях имеет преимущества, которыми обусловлен особый интерес к этому направлению: небольшая длина ключа и высокое быстродействие.

Эллиптической кривой над полем F будем называть гладкую кривую, задаваемую уравнением вида

$$Y^2 + a_1XY + a_3Y = X^3 + a_2X^2 + a_4X + a_6, a_i \in F. \quad (1)$$

Далее для произвольного поля будем использовать следующее определение: гладкая кривая, задаваемая уравнением вида (1), называется эллиптической кривой $E\varepsilon$ над полем F.

На множестве $E(F)$, состоящем из точек эллиптической кривой (1) и еще одного элемента – бесконечно удаленной точки, можно определить операцию, которая обладает свойствами операций группы, называемой абелевой [1, с. 90]. Уточним, что формально эта точка пока не являются точкой кривой. Принято рассматривать как аддитивную получающуюся при этом группу, а операцию называть операцией сложения и привычно обозначать знаком плюс. Упомянутая дополнительная точка выступает в качестве нейтрального элемента (при аддитивной записи – это ноль) этой группы и обозначается O.

Если K – расширение поля F, то $E(K)$ обозначает множество точек $(x, y) \in K^2$, удовлетворяющих (1), вместе с точкой O. Будем использовать обозначение $E(K)$ только в тех случаях, когда в рассмотрении участвует не только поле F, но и его расширение K.

По определению полагаем для любой точки $(x, y) \in E(F)$

Общая схема алгоритма сложения точек $P_1 = (x_1, y_1)$ и $P_2 = (x_2, y_2)$ в соответствии с определением операции сложения в группе точек эллиптической кривой представляется в следующем виде.

Эта страница представляет собой шаблон оформления текста доклада. Текст оформляется следующими стилями:

Алгоритм 1. Общая схема сложения и удвоения точек
ВХОД: Коэффициенты эллиптической кривой,
точки $P_1 = (x_1, y_1)$ (или $P_1 = O$) и $P_2 = (x_2, y_2)$ (или $P_2 = O$).
ВЫХОД: $P = P_1 + P_2$,
Вычислить: если $P_1 = O$, то $P = P_2$,
если $P_2 = O$, то $P = P_1$,
если $P_2 = -P_1$, то $P = O$,
если $x_1 \neq x_2$, то $P_2 = -(x_3, y_3)$,
иначе $P = 2P_1 = -(x_3, y_3)$.
Вернуть P.

В данный алгоритм включен случай удвоения точки (последняя строка), когда слагаемые не взаимно противоположны, но одинаковы. У некоторых кривых есть «самопротивоположные» точки (такие, что $P = -P$), правило удвоения к таким точкам не применимо, и в этом случае алгоритм, не доходя до последней строки, заканчивает работу.

Для группы точек эллиптической кривой проблема дискретного логарифма заключается в том, чтобы по известным точке P данной эллиптической кривой и точке $Q = kP$, определить числа k. Сложность этой проблемы не менее сложности DLP для поля, над которым определена кривая. Для эллиптических кривых неизвестны алгоритмы решения DLP, полиномиальные по сложности, но для суперсингулярных известен алгоритм сведения этой проблемы к DLP некоторого расширения невысокой степени поля, над которым определена кривая.

Классическая проблема Диффи – Хеллмана формулируется применительно к мультипликативной группе конечного поля. Она заключается в вычислении элемента α^{xy} по элементам α, α^x и α^y. Неизвестно, возможно ли ее решить не производя предварительные вычисления индексов x и y, т.е. минуя проблему дискретного логарифма. Говоря о циклической подгруппе группы точек эллиптической кривой, то применительно к ней эта проблема заключается в том, чтобы при известных точке $P \in E(F)$ и двух ее кратных k_AP и k_BP найти точку k_Ak_BP. Также неизвестно, можно ли это сделать без предварительного вычисления констант k_A и k_B, т.е. не решая DLP для эллиптических кривых.

Рассмотрим распределение ключей для классической криптосистемы (протокол Диффи – Хеллмана) [4, с. 102]. Стоит отметить, что в качестве ключа классической криптосистемы можно использовать неизвестную посторонним (секретную) случайную точку $(x, y) \neq O$ группы точек эллиптической кривой $E(F)$, если условиться, каким образом конвертировать ее в натуральное число, например, одну из координат, скажем, x считать двоичной записью натурального числа.

Модификацию протокола Диффи – Хеллмана можно использовать для получения такой секретной точки на двух терминалах открытого канала связи.

Допустим, что E – эллиптическая кривая и P – предварительно согласованная и опубликованная точка этой кривой. Абонент A выбирает, сохраняя в секрете случайное число k_A (секретный ключ A), вычисляет координаты точки k_AP (свою «половинку» ключа) и пересылает их абоненту В. Аналогично В выбирает секретный ключ k_B, вычисляет и пересылает абоненту A «половинку» k_BP ключа. Общим ключом является точка $P = k_Ak_BP$. A вычисляет ее, умножая на свой секретный ключ k_A на «половинку» ключа, вычисленную B, а B вычисляет эту же точку, умножая сообщение, поступившее от A, на свой секретный ключ k_B. Ввиду того, что группа точек эллиптической кривой абелева, результат не зависит от порядка вычисления и, следовательно, A и B имеют координаты секретной точки $k_A(k_BP) = k_B(k_AP) = k_Ak_BP = (x, y)$ и могут использовать x в качестве ключа симметричной криптосистемы (при условии достаточности длины этой двоичной записи, что зависит от порядка поля, над которым построена эллиптическая кривая, и также, если секретные ключи k_A и k_B были выбраны как случайные или как криптографически стойкие псевдослучайные числа). Теперь оба абонента A и B имеют одинаковые копии искомой секретной точки эллиптической кривой.

Проблема, с которой столкнется посторонний наблюдатель, имеющий намерение узнать секретный ключ, заключается в вычислении k_Ak_BP по известным P, k_AP, k_BP, но при неизвестных k_A, k_B, в этом и состоит проблема Диффи – Хеллмана для эллиптических кривых.

Кроме того, существует множество других протоколов для классической криптосистемы. Так, например, при помощи протокола Месси – Омуры можно произвести передачу сообщения от абонента A абоненту B по открытому каналу связи без предварительной передачи какой бы то ни было ключевой информации.

Первоначально протокол Месси – Омуры был описан применительно к мультипликативной группе Z^*p, где p – простое число. Он представляет собой аналог передачи секрета с помощью ящиков, запираемых на один или два замка. Так, абонент A запирает своим ключом ящик с письмом и пересылает ящик абоненту B, который запирает ящик своим ключом и отправляет его к A. Последний должен снять свой замок и вернуть ящик B, который снимает свой замок. Вместо механических замков абоненты (их может быть два или более) могут использовать электронные, т.е. хранящиеся в компьютерной памяти ключи. Описанный вариант системы возможно реализовать, не используя процедуры возведения в степень в конечных полях, однако задача дискретного логарифмирования достаточно сложна для В, чтобы тот по первому ключу от A не смог определить ключ и впредь читать не предназначавшиеся ему сообщения. Устойчивая система подписи сообщений является обязательным условием надежности. В случае, когда подписи не используются, любое постороннее лицо C может представиться абонентом B и вернуть A со своим ключом. А продолжит процедуру и, «сняв свой замок», откроет самозванцу C путь к расшифрованию сообщения. В связи с этим, шифрование вторым ключом должно сопровождаться аутентификацией.

Данный протокол в эллиптическом варианте предоставляет возможность передавать сообщение от A к B по открытому каналу без предварительной передачи какой-либо ключевой информации.

На основе рассмотренных выше протоколов можно сделать вывод, что основным преимуществом эллиптической криптографии является малый размер ключа относительно других схем асимметричного шифрования. Это свойство особенно важно при реализации криптографических протоколов в условиях ограниченности ресурсов памяти и производительности, например, при программировании смарт-карт.

С помощью эллиптических кривых можно зашифровать не только текст, но и фотографии. Задача заключается в том, чтобы изображение было наиболее неузнаваемым. Для этого будет использоваться стохастическое шифрование и «гаммирование».

«Гаммирование» заключается в сложении точек эллиптической кривой, соответствующих гамме, с соответствующими символам алфавита точками эллиптической кривой. «Гамма» – это псевдослучайная последовательность точек эллиптической кривой.

Первоначальные фотографии, которые следовало зашифровать с помощью описанного выше метода, представлены на рис. 1 в столбце «Оригинал изображения». При помощи различных коэффициентов получаем более или менее зашифрованные картинки. Во всех случаях первый режим не дает желаемого результата: изображения достаточно узнаваемы, и их передача по открытому каналу связи небезопасна, так как злоумышленник может увидеть скрытые на них данные. В случае использования второго режима удалось добиться стойкости алгоритма, и передача данных по открытому каналу связи полностью безопасна: зашифрованное изображение скрыто от посторонних.



Рис. 1. Пример алгоритма шифрования

На основании изучения данной работы можно выделить основные достоинства и недостатки эллиптической криптографии.

Гораздо меньшая длина ключа по сравнению с «классической» асимметричной криптографией, что повышает эффективность вычислений. Кроме того, скорость работы эллиптических алгоритмов гораздо выше, чем у классических. Это объясняется как размерами поля, так и применением более близкой для компьютеров структуры бинарного конечного поля.

Из-за маленькой длины ключа и высокой скорости работы, алгоритмы асимметричной криптографии на эллиптических кривых могут использоваться в смарт-картах и других устройствах с ограниченными вычислительными ресурсами.

Все плюсы эллиптической криптографии вытекают из одного конкретного факта: для задачи дискретного логарифмирования на эллиптических кривых не существует субэкспоненциальных алгоритмов решения, что позволяет уменьшить длину ключа и увеличить производительность. Однако если такие алгоритмы появятся, то это будет означать крах эллиптической криптографии.

Библиографический список

1. Болотов, А. А. Элементарное введение в эллиптическую криптографию. Алгебраические и алгоритмические основы / А. А. Болотов, С. Б. Гашков, А. Б. Фролов. – М. : КомКнига, 2006. – С. 89–98.

2. Соловьев, Ю. П. Эллиптические кривые и современные алгоритмы теории чисел / Ю. П. Соловьев, В. А. Садовничий, Е.Т. Шавгулидзе. – М. : Ижевск, 2003. – С. 60–92.

3. Хоффман, Л. Дж. Современные методы защиты информации / Л. Дж. Хоффман. – М. : Советское радио, 1980. – С. 87–164 с.

К. И. Никишин

Пензенский государственный университет, Пенза, Россия

ШИФРОВАНИЕ ИНФОРМАЦИИ С ИСПОЛЬЗОВАНИЕМ ПЕРЕСТАНОВОЧНОГО ШИФРА

Рассматривается программа шифрования/дешифрования текста с использованием перестановочного шифра. Перестановочный шифр разбивает информацию на блоки по некоторому количеству символов, для каждого такого блока осуществляется некоторая перестановка.

Информация является одним из признаков развития информационного общества [1]. В жизни каждого человека информация занимает особое место: личная информация, корпоративная информация и т.д. Сейчас трудно представить человека, который не пользуется сетью Интернет, не общается с другими пользователями в социальных сетях, по электронной почте.

Но любую информацию можно взломать, для обеспечения пользователям конфиденциальности предусмотрены различные механизмы шифрования. Само понятие шифрования включает в себя обратимое преобразование информации в целях скрытия от неавторизованных лиц, с предоставлением, в это же время, авторизованным пользователям доступа к ней.

В разных странах существуют различные стандарты шифрования данных, которые определены на законодательном уровне и регулируются нормативно-правовой базой страны. Самым распространенным алгоритмом является DES (Data Encryption Standart) [2]. Алгоритм Data Encryption Standart использует множество подстановок и перестановок, производит шифрование 64-битовых блоков с помощью ключа размером 64 бит. Российский стандарт шифрования данных представлен в ГОСТ 28147–89 «Система обработки информации. Защита криптографическая. Алгоритм криптографического преобразования» [3].

В статье рассматривается шифрование с перестановочным шифром. Данное шифрование предназначается для простых пользователей на уровне передачи информации через электронную почту, переписку в социальных сетях. Для крупных компаний данная программа не предназначается, поскольку необходимо зашифровывать информацию со стойкими алгоритмами шифрования.

Но исходя из результатов ежегодного исследования Norton Cybercrime Report за 2016 г. [4] совершения киберпреступлений следует, что средний ущерб от кибератаки на одного среднестатистического пользователя составляет $197, каждый пятый (21 %) онлайн-пользователь старше 18 лет становился жертвой кибератаки либо в социальных сетях, либо через мобильные устройства, либо через электронную почту, 39 % пользователей социальных сетей становились жертвами киберпреступлений.

Программа шифрования/дешифрования информации с использованием перестановочного шифра состоит из двух важных компонентов: шифратора и дешифратора программы. Интерфейс программы представлен на рис. 1.

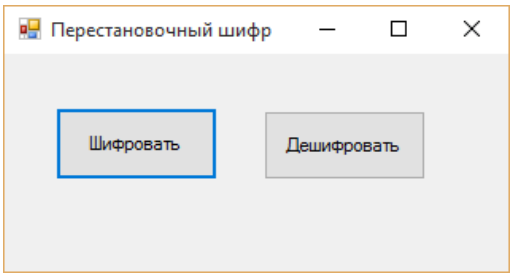


Рис. 1. Интерфейс программы шифрования

Перестановочный шифр с фиксированным периодом подразумевает разбиение исходной информации на блоки на некоторое количество символов и для каждого такого блока используется некоторая перестановка [5]. Ключом такого шифра является используемая при шифровании перестановочная матрица *P* или вектор *t*, указывающий правило перестановки. Таким образом, общее число возможных ключей определяется длиной блока *n* и равно *n!*. При дешифрации используется матрица обратной перестановки *D*, являющаяся обратной к матрице *P* по умножению, то есть $D*P = I$, где *I* – единичная матрица. Более подробно криптоанализ обсуждается в [6, 7].

Таким образом, при выполнении шифрования образуется таблица с заданной определенной длиной и горизонтально заполняется информацию, которую необходимо зашифровать. После этого информация считывается по столбцам таблицы (вертикально) и получается шифротекст (зашифрованная информация). Дешифрация информации уже осуществляется по полученному шифротексту и происходит восстановление исходной информации из шифротекста.

Программа шифратора информации работает следующим образом: пользователей указывается текстовый файл, который необходимо зашифровать. После этого программа считывает информацию из файла и заполняет новый массив информации с определенным размером блока информации. Обрабатывается массив согласно алгоритму перестановочного шифра, и шифротекст записывается в новый файл. Результат работы шифратора программы представлен на рис. 2.

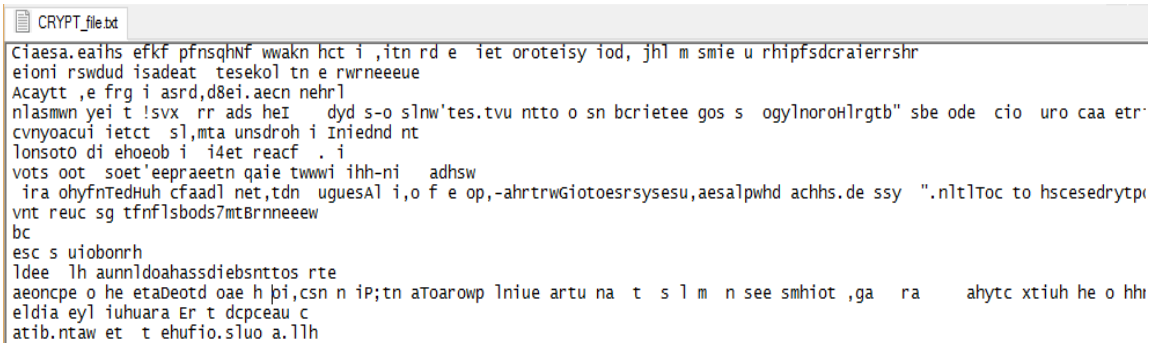


Рис. 2. Файл с зашифрованной информацией (шифротекст)

Программа дешифратора информации осуществляет обратные действия шифратора. Считывается информация уже из файла с шифротекстом, заполняется массив с определенным размером блока информации и восстанавливается исходная информация из шифротекста. Полученная информация сохраняется в файл.

Было проведено сравнение по содержимому исходного шифруемого файла с дешифрованным файлом, результат оказался положительным, поскольку должны были совпасть данные.

В статье рассмотрена реализация программы шифрования/дешифрования информации с использованием перестановочного шифра. Описана работа шифратора и дешифратора программы с учетом специфики перестановочного шифра.

Библиографический список

1. Понятие информационного общества. – URL: https://ru.wikipedia.org/wiki/Информационное_общество.
2. Стандарты шифрования данных. – URL: https://ru.wikipedia.org/wiki/Стандарты_шифрования_данных.
3. ГОСТ 28147–89 Системы обработки информации. Защита криптографическая. Алгоритм криптографического преобразования.
4. Исследования Norton Cybercrime Report. – 2016. – URL: https://us.norton.com/cybersecurity-insights
5. Шнайер, Б. Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си / Б. Шнайер. – М. : ТРИУМФ, 2002.
6. Sinkov, A. Elementary Cryptanalysis / A. Sinkov. – Mathematical Association of America, 1966.
7. Games, H. F. Cryptanalysis / H. F. Games. American Photographic Press. – 1937. (Reprinted by Dover Publications, 1956.1)

Д. В. Пащенко, Е. А. Бальзанникова

Пензенский государственный университет, Пенза, Россия

ПОВЫШЕНИЕ ТОЧНОСТИ ИДЕНТИФИКАЦИИ ПОЛЬЗОВАТЕЛЯ ПО БИОМЕТРИЧЕСКИМ ДАННЫМ КЛАВИАТУРНОГО ПОЧЕРКА

Рассмотрен вопрос защиты информационных систем от несанкционированного доступа с использованием биометрической информации. Рассматривается метод биометрической идентификации по клавиатурному почерку. Предложены и апробированы методы повышения точности идентификации.

На сегодняшний день защита конфиденциальной информации от несанкционированного доступа является крайне актуальной задачей. Для достижения этой цели используется комплекс средств, основанных на одном или нескольких компонентах:

- атрибуты – к ним относятся пропуск, ключ, proxy-карта, документы;
- знания, такие как пароли, коды или ответ на вопрос;
- биометрическая информация – совокупность индивидуальных и уникальных физиологических параметров пользователя.

Однако системы, обеспечивающие безопасность данных на основании собственности или данных, слишком уязвимы и могут легко пострадать от подделки, воровства и других факторов. Поэтому все больший интерес вызывают методы биометрической идентификации, позволяющие определить личность человека по его физиологическим характеристикам путем распознания по заранее сохраненным образцам. Подобные способы аутентификации удобны для пользователей тем, что носители информации находятся всегда при них, не могут быть утеряны либо украдены. Биометрический контроль доступа считается более надежным, так как идентификаторы не могут быть переданы третьим лицам, скопированы.

Основными методами, использующими статические биометрические характеристики человека, являются идентификация по папиллярному рисунку на пальцах, радужной оболочке, геометрии лица, сетчатке глаза, рисунку вен руки, геометрии рук. Также существует семейство методов, использующих динамические характеристики: идентификация по голосу, динамике рукописного подчерка, сердечному ритму, походке. Таким образом, организация защиты системы от несанкционированного доступа (НСД) с использованием средств биометрической идентификации, требует наличия дополнительного оборудования.

Исходя из этого, как правило, системы аутентификации по биометрическим данным размещаются на объектах с повышенными требованиями по безопасности и защиты от НСД, тогда как дороговизна такого оборудования или неудобство ежедневного использования являются существенным ограничением широкого применения подобных систем.

Однако, на сегодняшний день существуют методы и средства идентификации и аутентификации пользователей при помощи стандартных устройств ввода персональной ЭВМ: клавиатуры и мыши. Механика набора текста на клавиатуре, по сравнению с особенностями движения мыши, обладает более простыми и легко интерпретируемыми параметрами, такие как время удержания клавиши или интервал между нажатиями клавиш, исходя из чего, в качестве биометрических данных в данной работе предлагается использовать именно клавиатурный почерк. Подобно рукописному почерку, клавиатурный почерк является уникальными для каждого человека, что позволяют с высокой долей вероятности идентифицировать пользователя, известного системе или классифицировать пользователя как «чужого», если он не соответствует ни одному образу, имеющемуся в базе. Зачастую, ввиду особенностей такой биометрической информации, системы анализа клавиатурного почерка обладают достаточно высокой точностью, по сравнению с системами, основанных на физиологических параметрах.

Определим клавиатурный почерк как совокупность параметров нажатий клавиш, отражающих процесс набора контрольной фразы, которая является результатом сознательных и бессознательных процессов мышления, а также конструктивных особенностей клавиатуры.

Существует множество эффективных и надежных алгоритмов идентификации пользователя по образу клавиатурному почерка. Однако имеется общая проблема данного вида идентификации: ухудшение качества идентификации ввиду нестабильности клавиатурного почерка пользователя. Даже у опытных пользователей персонального компьютера с формировавшейся манерой и механикой набора текста на клавиатуре могут наблюдаться существенные отклонения в динамике набора одного и того же текста. Эти отклонения могут быть обусловлены различными факторами: психофизическое состояние, вариации последовательности используемых пальцев при наборе на клавиатуре в рамках устоявшегося почерка или случайные или независящие от пользователя отклонения. Такого рода отклонения имеют особую важность, так как они повлияют на формирование эталонного образа клавиатурного почерка пользователя и могут в дальнейшем негативно сказаться в дальнейшем на точности идентификации.

Традиционно для повышения точности результатов анализа данных применяется рад мер по предварительной обработке данных, которые включают в себя: фильтрацию выбросов и нормализацию данных. Также возможно извлечение новых, линейно независимых признаков из входных данных.

Для проведения оценки точности алгоритмов были собраны данные о вводе парольной фразы несколькими пользователями, для которых методом кросс-валидации была произведена вероятности ложного срабатывания (FAR) и ложного отказа (FRR) для двух пользователей, которые условно были определены как образцы стабильного и нестабильного почерка.

Результаты данного анализа для различных порогов доступа показаны на рис. 1. На графиках слева показаны результаты для данных без предварительной обработки, справа – для данных после проведения фильтрации и нормализации данных. Графики сверху отражают результаты для образцов «нестабильного».

Как видно из приведенных результатов, предварительная обработка позволяет добиться повышения качества идентификации только для «стабильного» почерка, для «нестабильного» улучшения несущественны.

Таким образом, основной проблемой применения алгоритмов идентификации пользователя по клавиатурному почерку является нестабильность основных характеристик, обусловленная вариациями способов набора одной и той же парольной фразы в рамках клавиатурного почерка одного пользователя.

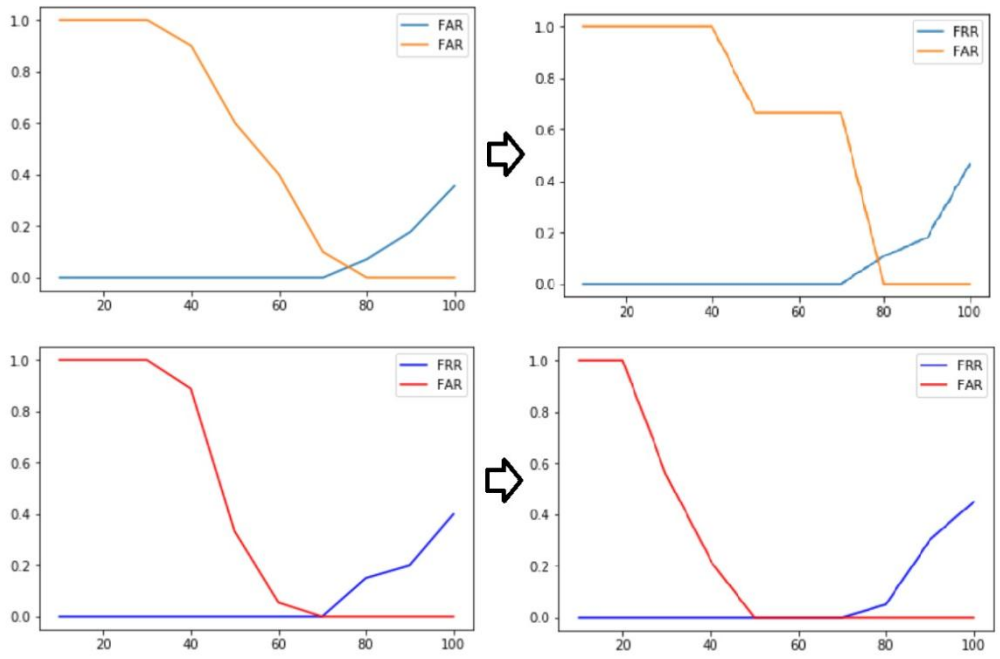


Рис. 1. Вероятность ошибок FAR и FRR для различных порогов допуска
с применением предварительной обработки

В большинстве случаев набор значений одного признака рассматривается как единая выборка, подчиняющаяся некоторому закону распределения. В рамках данной работы предлагается рассматривать значения признаков в виде отдельных групп, соответствующих альтернативными вариантами способа набора парольной фразы или текста. Предполагаем, каждая из групп также подчиняется нормальному закону распределения. Таким образом, данный способ обработки предполагает предварительную кластеризацию каждого набора признаков и последующим применением алгоритма идентификации применительно для каждого кластера.

Результаты оценки качества идентификации с применением кластеризации и предварительной обработки представлены на рис. 2.

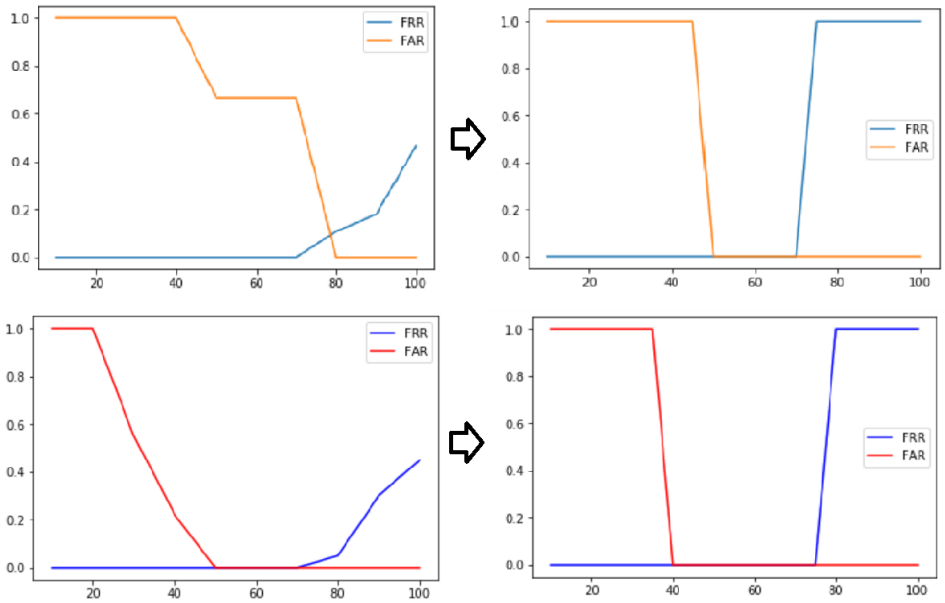


Рис. 2. Вероятность ошибок FAR и FRR для различных порогов допуска
с применением кластеризации

Таким образом, применение предложенного метода повышения качества идентификации по клавиатурному почерку с использованием кластеризации позволяет компенсировать основной недостаток всех существующих методов – снижение точности распознавания для пользователей с нестабильной механикой набора парольной фразы. При этом за счет значительного расширения возможностей существующих алгоритмов, возможно более широкое распространение методов идентификации пользователей по биометрическим образам клавиатурного почерка.

Библиографический список

1. Иванов, А. И. Биометрическая идентификация личности по динамике подсознательных движений / А. И. Иванов. – Пенза : Изд-во ПГУ, 2000. – 188 с.

2. Брюхомицкий, Ю. А. Статистические методы распознавания клавиатурного почерка / Ю. А. Брюхомицкий // Известия Южного федерального университета. Технические науки. Тематический выпуск. – 2010. – С 139–147.

3. Ходашинский, И. А. Технология усиленной аутентификации пользователей информационных процессов – Управление, вычислительная техника и информатика / И. А. Ходашинский, М. В. Савчук, И. В. Горбунов, Р. В. Мещеряков // Доклады ТУСУРа. – 2011. – № 2 (24), ч. 3. – С. 236–248.

4. Брюхомицкий, Ю. А. Гистограммный метод распознавания клавиатурного почерка / Ю. А. Брюхомицкий // Известия Южного федерального университета. Технические науки. – 2010. – № 11, т. 112. – 8 с.

5. Воронцов, К. В. Машинное обучение : курс лекций / К. В. Воронцов. – 2017. – URL: www.machinelearning.ru

Н. П. Вашкевич, П. Ю. Мясин, С. Ю. Поздняков

Пензенский государственный университет, Пенза, Россия

ПРИНЦИПЫ ПОСТРОЕНИЯ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ В УСЛОВИЯХ СОВРЕМЕННОГО ИНФОРМАЦИОННОГО ПРОТИВОБОРСТВА

Рассмотрены некоторые информационные угрозы автоматизированных систем управления военного назначения, их жизненный цикл. Рассматриваются элементы и комплексы средств защиты информации, а также альтернативный способ представления уязвимостей программного обеспечения.

Эффективная деятельность любых структур военного назначения невозможна без использования современных информационных технологий, что предъявляет ряд требований к обеспечению информационной безопасности.

При использовании современных информационных технологий (ИТ) в автоматизированных системах управления (АСУ) военного назначения, таких, например, как компьютерные сети и базы данных, ценность информации (с точки зрения решения задач ведения боевых действий) увеличивается, поскольку она дает возможность повысить степень осведомленности, улучшить взаимодействие между командованиями различного уровня, органами военного управления и разведки и, тем самым, реализовать свое информационное превосходство [1]. Сравнивая требования информационной безопасности, применяемые в структурах военного назначения стран НАТО, одним из ключевых элементов информационного противоборства является киберкомандования родов вооруженных сил.

Например, комитет начальников штабов вооруженных сил США утвердил документ «Информационные операции», в котором излагались взгляды американского военно-

го руководства на их подготовку и проведение, уточнены цели, задачи основные принципы информационного противоборства, а так же обязанности должностных лиц по подготовке и проведению таких операций в мирное и военное время[2].

Как следовало из документа, информационные операции представляют собой комплекс мероприятий по воздействию на людские и материальные ресурсы противника для того, чтобы затруднить или сделать невозможным принятие верного решения с одновременной защитой своих информационно – коммуникационных сетей и компьютерных систем. Такие операции включали в себя пять основных составляющих:

- радиоэлектронную борьбу (electronic warfare);
- психологические операции (psychological operations);
- операции в информационно – коммуникационных сетях (computer network operations);
- военную дезинформацию (military deception).

Рассмотрим одну из информационных угроз – уязвимость программного обеспечения (ПО). Уязвимость (информационных систем); брешь – свойство информационной системы (ИС), обуславливающее возможность реализации угроз безопасности обрабатываемой в ней информации [3].

Наличие в ИС уязвимостей позволяет злоумышленникам проводить деструктивные воздействия на нее, которые реализуются с помощью эксплойтов определенной последовательности команд целевого процессора, которые используют уязвимость ИС. Процесс применения разработанного эксплойта называют атакой. Типовой жизненный цикл уязвимости ПО [5] приведен на рис. 1.

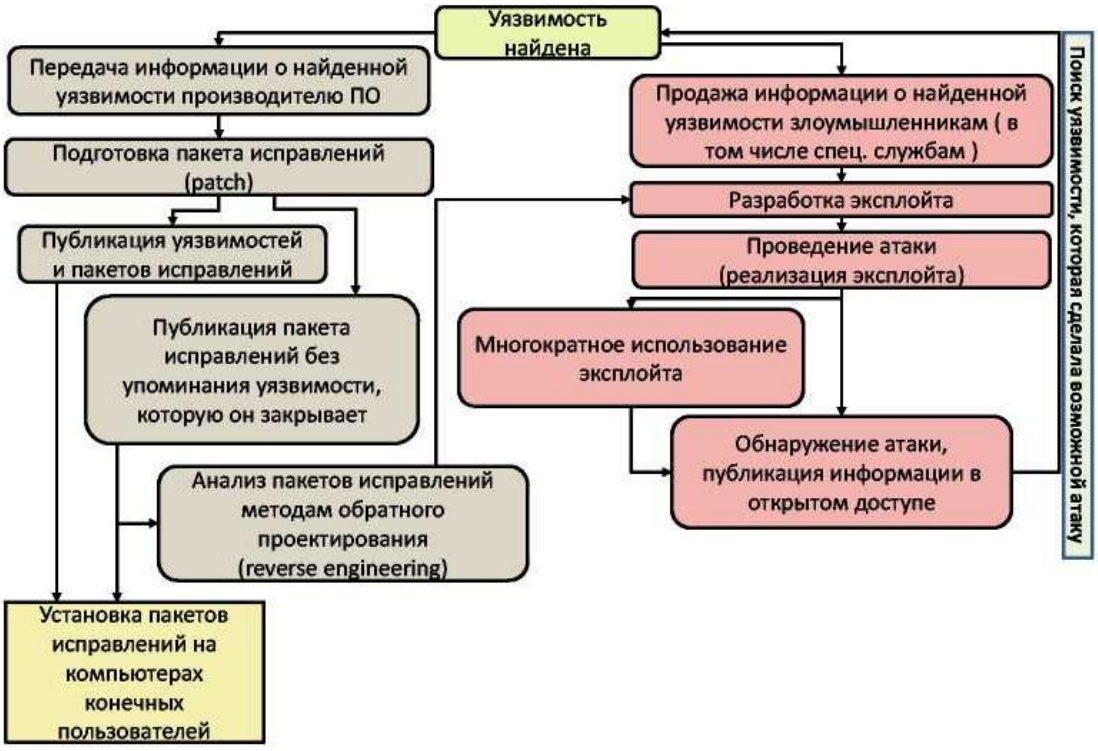


Рис. 1. Типовой жизненный цикл уязвимости ПО

Одним из выходов предотвращения информационных угроз является создания централизованной системы управления военного назначения.

Централизованная система управления военного назначения должна охватывать как унаследованные, хорошо себя зарекомендовавшие в условиях реальной эксплуатации, тех-

нические решения, так и перспективные, разрабатываемые в настоящее время элементы и комплексы средств защиты информации.

В качестве некоторых элементов и комплексов средств защиты информации рассмотрим:

- унифицированные аппаратно-программные модули доверенной загрузки с функциями управления по сети информационно-управляющей системы военного назначения (ИУС ВН), обеспечения доверенной виртуализации и контроля отчуждаемых носителей информации;
- программно-аппаратные комплексы за щиты информации от информационно технического воздействия;
- комплексы защиты информации технологии «тонкий клиент» и средств виртуализации;
- высокопроизводительные криптографические средства защиты информации, включая криптографические пулы;
- программные комплексы централизованного управления средствами защиты информации взаимодействующих ИУС ВН в рамках одной мультиплатформенной системы управления информационной безопасностью (СУИБ);
- средства защиты от НСД к вычислительным ресурсам центров обработки данных (ЦОД) и средства оперативной миграции виртуальных машин и серверов;
- автоматизированные средства построения ложных объектов;
- средства взаимодействия с несколькими удостоверяющими центрами;
- средства централизованной биометрической идентификации и аутентификации.

Современные тенденции развития ИУС ВН требуют при создании АСУ ВН учитывать следующие перспективные направления развития систем комплексной защиты информации;

- разработку средств защиты информации распределенных вычислений и услуг ЦОД («облачные вычисления»);
- реализацию криптографических сервисов для распределенных вычислений;
- обеспечение централизованного управления инфраструктурой электронной подписи с учетом построения доверенного пространства многоуровневой системы удостоверяющих центров;
- реализацию технологии интеграции АС, основанной на интеграции сервисов с предоставлением возможностей виртуализации выносных рабочих мест, серверных компонентов и электронного документооборота, включающего обработку потоков документов, заданий, индексацию и поиск информации (Интеграционный комплекс информационно-сетевых сервисов (ИКИСС);
- реализацию автоматизированных средств (сервисов) оценки эффективности защиты взаимодействующих ИУС ВН;
- защиту информации, обрабатываемой средствами виртуализации, в том числе с использованием виртуальных средств организации взаимодействия между сегментами АСУ (виртуальная сетевая инфраструктура, рабочие места и вычислительные комплексы);
- централизованное управление персональными биометрическими данными для доступа к ресурсам АС, а так же на объекты охраны;
- реализацию возможностей централизованного управления гетерогенными СЗИ взаимодействующих ИУС ВН в рамках единой СУИБ;
- построение системы компетенций СЗИ и системы взаимного доверия сервисов взаимодействующих ИУС ВН;
- криптографическую защиту информации при передаче по беспроводным каналам связи, включая мобильные устройства и оконечные исполнительные устройства;
- внедрение перспективных средств идентификации абонентов и носителей ключевой информации и др.

Таким образом, только на основе реализации комплексного подхода в обеспечении отражения информационных угроз на основе перечисленных технологий и механизмов защиты может быть достигнут требуемый высокий уровень информационной безопасности,

надежности и функциональной устойчивости АСУ ВН в условиях современного информационного противоборства.

Библиографический список

1. Зубарев, И. В. Кибер безопасность автоматизированных системы управления военного назначения / И. В. Зубарев, И. В. Жидков, И. В. Кадушкин // Вопросы кибербезопасности. – 2013. – № 1.

2. Information Operations Primer: Fundamentals of Information Operations. – U.S. Army War College, 2011. – 204 p.

3. ГОСТР 50922–2006 Защита информации. Основные термины и определения.

4. Engle, S. Tree Approach to Vulnerability Classification / S. Engle, S. Whalen, D. Howard, M. Bishop // Technical Report CSE-2006-10, Dept. of Computer Science, University of California at Davis, Davis, CA 95616-8562.

5. Марков, А. С., Систематика уязвимостей и дефектов безопасности программных ресурсов / А. С. Марков, А. А. Фадич // Защита информации: INSIDE. – 2013. – № 3. – С. 18–21.

6. Вашкевич, Н. П. Недетерминированные автоматы в проектировании систем параллельной обработки : учеб. пособие / Н. П. Вашкевич. – Пенза : Изд-во ПГУ, 2004. – 280 с.

7. Бородакий, Ю. В. Предложения в межведомственную концепцию создания доверенной аппаратно-программной среды для автоматизированных систем органов управления / Ю. В. Бородакий, А. Ю. Добродеев, И. В. Бутусов // Вопросы радиоэлектроники. – 2013. – Т. 3, № 2. – С. 15–22.

8. Бородакий, Ю. В. Основные задачи и проблемы создания криптографической подсистемы защиты распределенных автоматизированных систем управления и связи специального назначения / Ю. В. Бородакий, А. Ю. Добродеев, Ю. П. Свиридюк, П. А. Нащекин // Информационное противодействие угрозам терроризма. – 2005. – № 4. – С. 176–178.

А. А. Яужев

Пензенский государственный университет, Пенза, Россия

КОМБИНИРОВАННОЕ СРЕДСТВО АУТЕНТИФИКАЦИИ

Использование аппаратных средств аутентификации в многопользовательских информационных системах, не реализующих поддержку специализированных токенов, для ускорения и упрощения процедуры аутентификации по паролю или пин-коду.

Информационные системы выполняют разграничение доступа к своим подсистемам на основе данных, предоставляемых пользователями в процессе аутентификации. Эти данные вне зависимости от формы их представления представляют собой факторы аутентификации [1], присущие конкретному пользователю. Факторы аутентификации позволяют однозначно идентифицировать пользователя и предоставить ему некоторый набор прав.

К факторам аутентификации относят:

- токены – специальные устройства и их программные эмуляторы, формирующие аутентификационные последовательности на основе некоторого секрета с использованием определенного алгоритма (например, TOTP);
- биометрические данные – физические особенности пользователя, преобразуемые к цифровому виду (отпечаток пальца, рис. радужной оболочки);
- некоторые сведения, известные только пользователю, представленные в текстовом виде – пароль или пин-код.

Последовательность символов в сочетании с именем пользователя (логином) является наиболее простым в реализации способом аутентификации пользователя в информационной системе. Разработчику, а также пользователю, не требуется использовать специализированное программное или аппаратное обеспечение для разработки и работы

в системе соответственно. Однако для обеспечения защиты системы от несанкционированного доступа с использованием существующих механизмов аутентификации администраторы системы могут устанавливать особые требования к длине, алфавиту, а также сроку действия паролей – эти меры могут создавать существенные сложности для пользователей, а в некоторых случаях – снизить уровень безопасности.

Для решения проблемы хранения множества паролей к множеству информационных систем были разработаны менеджеры паролей – специализированные программы, хранящие аутентификационные данные в зашифрованных базах данных. Такой подход имеет несколько существенных недостатков: пользователь должен быть авторизован в операционной системе, а также иметь соответствующие права для запуска программы.

На основе микроконтроллера ATMega32U4, а также с использованием средств языка программирования C++ разработано устройство, предоставляющее аутентификационные данные (логины, пароли и пин-коды) для заданного множества информационных систем. Устройство представляет собой USB-совместимый токен, реализующий базовые функции менеджера паролей.

Устройство представляет собой изолированное хранилище данных, используемых для формирования аутентификационных данных той или иной информационной системы. Формирование паролей осуществляется с использованием шума, сформированного функцией губки Keccak [2] на основе данных контекста информационной системы.

В состав устройства входят:

- микроконтроллер Microchip ATMega32U4;
- подсистема ввода-вывода: клавиши управления и экран на основе контроллера SSD1306, сопряженный с микроконтроллером по шине I2C;
- подсистема хранения данных в виде выделенной микросхемы памяти серии Atmel AT24C;
- регулятор напряжения для обеспечения электрической совместимости компонентов.

Контекст информационной системы – это данные, на основе которых из фиксированного символьного алфавита формируются аутентификационные данные (секрет). В контекст входят: наименование системы, наименование формируемых данных, а также один из счетчиков. Процесс формирования секрета представлен на рис. 1.

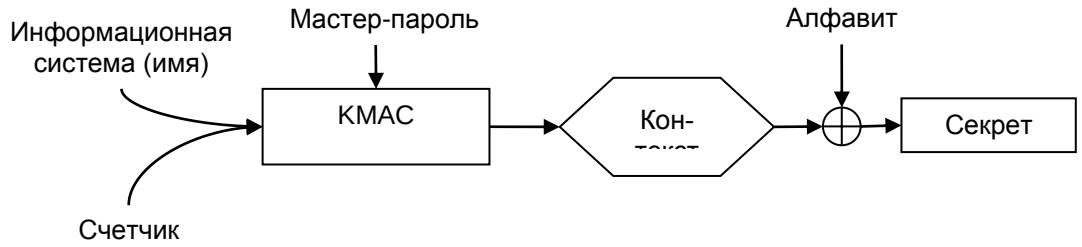


Рис. 1. Формирование секрета

Все хранимые данные подвержены маскированию с использованием мастер-ключа пользователя; при использовании демаскированных данных осуществляется их контроль целостности, а также применяются меры по маскированию используемых буферов в памяти микроконтроллера.

Устройство использует эмуляцию клавиатуры для вывода данных в компьютер, к которому оно подключено. Это достигается с использованием библиотеки LUFA, реализующей упрощенный стек протокола USB на микроконтроллерах AVR [3]. Взаимодействие осуществляется в одностороннем порядке (только вывод), за исключением этапа начальной инициализации, в ходе которого пользователь вводит список информационных систем и мастер-ключ.

Криптографические операции, выполняемые устройством, основаны на функции Keccak, используемой в различных режимах: формирование псевдослучайной последовательности (маскирование), хеширование и MAC (обеспечение контроля целостности).

Устройство может быть использовано в существующих информационных системах, использующих пару логин-пароль в качестве фактора аутентификации. Для его работы в среде операционных систем Windows, Linux и Android не требуется использование специализированных драйверов: используется стандартный драйвер клавиатуры. Перед промышленным внедрением необходимо провести опытную эксплуатацию и убедиться в полной совместимости с конкретной информационной системой.

Библиографический список

1. Смит, Р. Аутентификация: от паролей до открытых ключей / Р. Смит. – М. : Вильямс, 2002. – С. 432.

2. Bertoni, G. Sponge Functions / G. Bertoni, J. Daemen, M. Peeters, G. Van Assche. – URL: http://sponge.noekeon.org/SpongeFunctions.pdf (дата обращения: 19.07.2017)

3. Баранов, В. И. Применение микроконтролеров AVR. Схемы. Алгоритмы. Программы / В. И. Баранов. – М. : Додэка XXI, 2006. – С. 288.

ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ И РАСПОЗНАВАНИЕ ОБРАЗОВ

Г. Ч. Шушкевич, С. В. Шушкевич

Гродненский государственный университе им. Янки Купалы, Гродно, Беларусь

ВИЗУАЛИЗАЦИЯ РЕШЕНИЙ ПРИКЛАДНЫХ ЗАДАЧ С ИСПОЛЬЗОВАНИЕМ ОБЛАЧНЫХ ТЕХНОЛОГИЙ WOLFRAM CLOUD

Обсуждается применение Wolfram Cloud для визуализации решений прикладных задач, описываемых дифференциальными уравнениями.

Применение компьютерной математики существенно расширяет возможности автоматизации всех этапов компьютерного моделирования [1, 2]. Системы компьютерной математики (СКМ) позволяют провести исследование проблемы, анализ данных, моделирование, тестирование, проверку существования решения, оптимизацию, документирование и оформление результатов, они позволяют сосредоточить основное внимание на сущности проблемы, оставляя в стороне технику классической математики, детали вычислительных методов и алгоритмических процедур, нюансы языков программирования и команд операционной системы.

В последние годы разработчики СКМ стали активно переходить на online-сервисы, что обусловлено следующими причинами:

– коммерческие СКМ достаточно дорогие и не каждый пользователь может приобрести программы с высокой стоимостью индивидуальной лицензии, а покупка недорогой подписки на веб-сервис выглядит намного привлекательнее, что позволяет компаниям расширить пользовательскую базу и увеличить прибыль;

– поддержка кроссплатформенности. Переход к web-сервису – это самый простой путь переноса функциональности СКМ на все существующие платформы, включая мобильные;

– мода на «облака́» и web-ориентированность.

Облачные вычисления содержат специализированный спектр технологий обработки и передачи данных и представляют собой динамически масштабируемый способ доступа к внешним вычислительным ресурсам в виде сервиса, предоставляемого посредством Интернета, при этом пользователю не требуется никаких особых знаний об инфраструктуре «облака» или навыков управления этой «облачной» технологией [3-5].

Можно выделить несколько основных технологий этого направления:

– программное обеспечение как сервис (Software as a Service, SaaS);

– платформа как сервис (Platform as a Service, PaaS);

– инфраструктура как сервис (Infrastructure as a Service, IaaS);

– данные как сервис (Data as a Service, DaaS);

– рабочее место как сервис (Workplace as a Service, WaaS);

– все как сервис (All as a Service, AaaS).

Software as a Service (SaaS) можно рассматривать как наиболее интересный для образования и научных исследований сервис облачных технологий [5]. С помощью SaaS пользователь получает бесплатный или за небольшую плату онлайн-сервис, предоставляющий полные или ограниченные функциональные возможности требуемого программного продукта. Среди СКМ подобный сервис представлен, например, для Maple, Mathcad, Mathematica, Scilab обеспечивая доступ не только к справочной информации, но и возможности проведения расчетов и хранения данных.

Использованию облачных технологий способствует сама программная архитектура СКМ. Все они состоят из аналитического ядра, которое является интерпретатором командного языка СКМ, и оболочки, обеспечивающей взаимодействие с пользователем, постро-

ние графиков или отображение формул в привычном виде. Такую архитектуру несложно преобразовать в облачный сервис – достаточно создать оболочку в виде web-приложения и запустить ядро на сервере.

Рассмотрим примеры применения облачного сервиса Wolfram Cloud [5], основанного на языке Wolfram Language, для решения некоторых прикладных примеров.

Задача 1. Рассмотрим электрический контур с последовательно соединенными элементами: сопротивлением R=1000 Ом, индуктивностью L=1 Г и емкостью C=6,25*10 $^{-6}$ Ф. Начальный заряд в контуре равен нулю. В момент времени t=0 к нему прикладывается постоянная э.д.с, равная 24 В. При t=0.001 с величина тока равна 0,031 А. Построить график тока i(t) и определить величину тока в момент времени t=0.

Решение данной задачи сводится к решению граничной задачи для заряда Q(t):

$$\frac{d^2}{dt^2}Q(t) + 1000\frac{d}{dt}Q(t) + 16*10^4 Q(t) = 24,$$

$$Q(0) = 0,\ \ i(0.001) = \frac{d}{dt}Q(0.001) = 0,031.$$

Mathematica–документ.

1. Решение граничной задачи, нахождение функции тока i(t).

```
In[1]:= Solution = DSolve[{Q''[t] + 1000 * Q'[t] + Q[t] * 10 ^ 6 / 6.25 == 24,
          Q[0] == 0, Q'[0.001] == 0.031},
         Q[t], t] // Simplify // Flatten
Out[1]= {Q[t] → 0.00015 - 0.0000328948 e^(-800. t) - 0.000117105 e^(-200. t)}

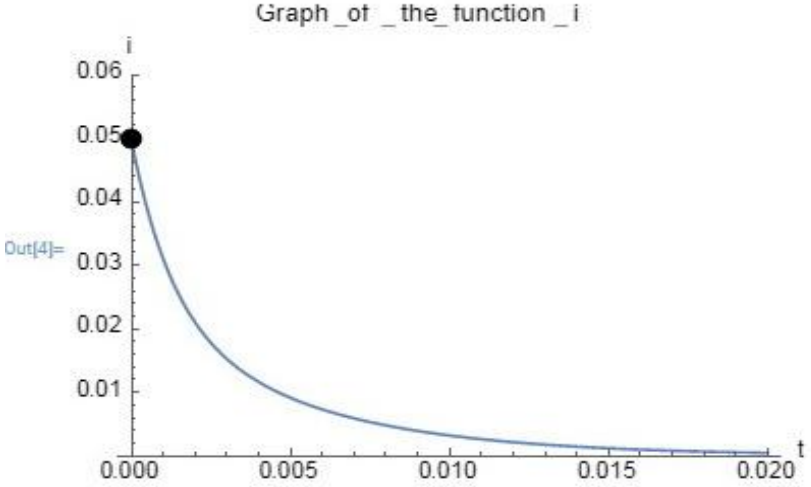
In[2]:= Qs[t_] = Q[t] /. Solution[[1]];

In[3]:= is[t_] = ∂t Qs[t];
```

2. Построение графика функции i(t), вычисление i(0) (значение изображено на графике).

```
In[4]:= Plot[is[t], {t, 0, 0.02}, AxesOrigin → {0, 0},
          PlotRange → {0, 0.06}, Epilog → {PointSize[0.03],
            Point[{0, is[0]}]},
          AxesLabel → {HoldForm[t], HoldForm[i]},
          PlotLabel → HoldForm[Graph _of _the_ function _i ],
          LabelStyle → {FontFamily → "Times New Roman", 12,
            GrayLevel[0], Bold}]
```

Out[4]=



Задача 2. Механическая система состоит из тела массы m, связанного с жесткой стенкой через пружину постоянной жесткости k, демпфером с коэффициентом демпфирования с и пружиной с нелинейной характеристикой, создающей восстанавливающую силу, равную произведению постоянной p на смещение в третьей степени.

Построить анимацию смещения $x(t)$, $0 \le t \le 1$, для значений коэффициента $0 \le c \le 1\ \text{Н}\cdot\text{c/cm}$ с шагом 0,1, если $m=0{,}01\ \text{Н}\cdot\text{c}^2/\text{cm}$, $k=2\ \text{Н/cm}$, $p=2\ \text{Н/cm}^3$.

Решение данной задачи сводится к решению дифференциального уравнения:

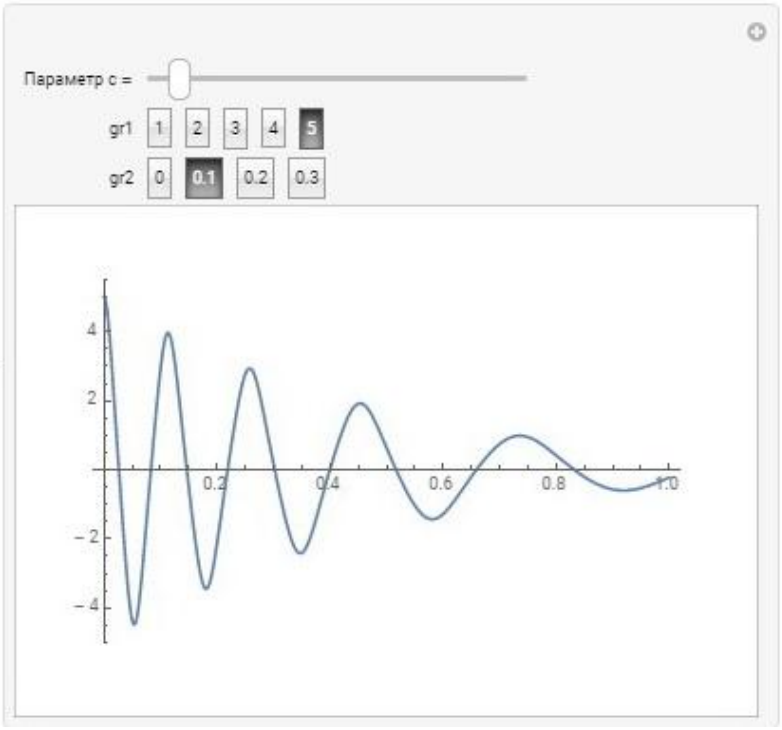
$$m\frac{d^2}{dt^2}x(t)+c\frac{d}{dt}x(t)+kx(t)+px^3(t)=0,\ x(0)=gr1,\quad \frac{d}{dt}x(0)=gr2.$$

Mathematica–документ.

1. Решение численное решение задачи.

```
In[1]:= Manipulate[Plot[Evaluate[x[t]/. First[NDSolve[ {0.01*x''[t1]+c*x'[t1]+
          2*x[t1]+2*x[t1]^3==0,
          x[0]==gr1, x'[0]==gr2}, x, {t1, 0, 1}]]], {t, 0, 1}, ImagePadding→25,
   PlotRange→Automatic],
  {{c, 1, "Параметр c = ", TraditionalForm[c]}, 0, 1, 0.01},
  {gr1, {1, 2, 3, 4, 5}, ControlType→SetterBar}, {gr2, {0, 0.1, 0.2, 0.3},
   ControlType→SetterBar}]
```

2. Анимация решения в зависимости от выбранных граничных условий.



Библиографический список

1. Самарский, А. А. Математическое моделирование в информационную эпоху /А. А. Самарский, А. П. Михайлов // Вестник Российской Академии наук. – 2004. – Т. 7, № 9. – С. 781–784.

2. Шушкевич, Г. Ч. Динамическая визуализации численных решений дифференциальных уравнений в системе Mathcad / Г. Ч. Шушкевич, С. В. Шушкевич // Информатизация инженерного образования : тр. Междунар. науч.-практ. конф. ИНФОРИНО–2016 (г. Москва, 12–13 апреля 2016 г.). – М. : Изд. дом МЭИ, 2016. – С. 477–480.

3. Облачные вычисления и сервисы на базе облачных вычислений. – URL: http://www.lessons-tva.info/archive/nov031.html (дата обращения: 10.08.2017).

4. Шушкевич, Г. Ч. Облачные вычисления в Wolfram Mathematica / Г. Ч. Шушкевич, С. В. Шушкевич // Инновационные технологии в современном образовании : сб. тр. по материалам III Междунар. науч.-практ. интернет-конф. (Наукоград Королев, 18 декабря 2015 г.). – М. : Научный консультант, 2016. – С. 771–774.

5. Тарнавский, Г. А. Технология облачных вычислений в математическом моделировании / Г. А. Тарнавский // Инф. тех. и выч. систем. – 2010. – № 4. – С. 66–76.

6. Wolfram Cloud. – URL: http://www.wolframcloud.com/ (дата обращения: 12.08.2017).

А. Ф. Каперко, В. П. Кулагин

Национальный исследовательский университет «Высшая школа экономики», Москва, Россия

ИНТЕЛЛЕКТУАЛЬНЫЕ МЕТОДЫ КОНТРОЛЯ ИОНИЗИРУЮЩЕГО ИЗЛУЧЕНИЯ НА ОСНОВЕ СПЕКТРОМЕТРА С АЛМАЗНЫМИ ДЕТЕКТОРАМИ

Рассматривается использование в качестве интеллектуального метода контроля потоков ионизирующего излучения нейросетевой подход обработки выходной информации со спектрометра, построенного на алмазных детекторах. С помощью спектрометра анализируются 24 выходных сигнала, содержащих интегральные количественные характеристики потоков ионизирующего излучения. Обрабатывается информация о потоках электронов, протонов и тяжелых заряженных частиц в энергетическом диапазоне от 0 до 10000 МэВ. Предложен математический аппарат, позволяющий получать информацию о дифференцированных значениях плотностей потоков ионизирующего излучения в 21 энергетическом диапазоне. Модель преобразования информации в тракте регистрации спектрометра основывается на использовании алгоритма моделирования в программном пакете GEANT4.

Современные устройства контроля параметров потоков ионизирующего излучения разрабатываются для эксплуатации в критических условиях. Создание перспективных детекторов потоков ионизирующего излучения необходимо для применения в системах контроля технологических параметров и диагностики оборудования ядерных энергетических установок, а также для использования в спектрометрах космического применения.

Для обеспечения длительного функционирования аппаратуры при интенсивном радиационном воздействии, особенно таких факторов, как потоки тяжелых заряженных частиц, электронов и протонов обычно используются алмазные детекторы. В качестве обоснования выбора алмазных детекторов для регистрации ионизирующего излучения, а в особенности космического излучения, наиболее значимыми параметрами являются следующие:

- радиационная и химическая стойкость;
- высокая теплопроводность;
- широкая запрещенная зона – 5.5eV, что обеспечивает низкие токи утечки и низкий уровень шума;
- высокая чувствительность, что определяет небольшие размеры и высокое разрешение детектора.

Для контроля потоков ионизирующего излучения предлагается использовать спектрометр с пятью алмазными детекторами [1]. При попадании ионизирующей частицы излучения в алмазный детектор в нем возникает электрический заряд Q, пропорциональный энергии E, потерянной частицей в детекторе. Выходной сигнал каждого детектора усили-

вается с помощью отдельного зарядочувствительного усилителя, на выходе которого появляются импульсы напряжения с амплитудой, пропорциональной заряду, а, следовательно, энергии, переданной частицей детектору. Выходные сигналы зарядочувствительных усилителей с помощью аналого-цифрового преобразования сортируются по амплитудным уровням на несколько диапазонов. В цифровой схеме спектрометра потоки ионизирующего излучения регистрируются с помощью 24-х счетчиков частиц, сигналы от которых превысили соответствующие амплитудные уровни.

Перед каждым алмазным детектором установлен селективный фильтр в виде металлической пластины с заданной толщиной рис. 1. Проходя через материал фильтра, ионизирующая частица теряет часть своей энергии, что приводит к смещению спектра поглощения энергии в алмазном детекторе. В результате, за время цикла измерения и контроля в 24 счетчиках частиц спектрометра фиксируются данные о количестве частиц ионизирующего излучения, прошедших через спектрометр из соответствующего энергетического диапазона.

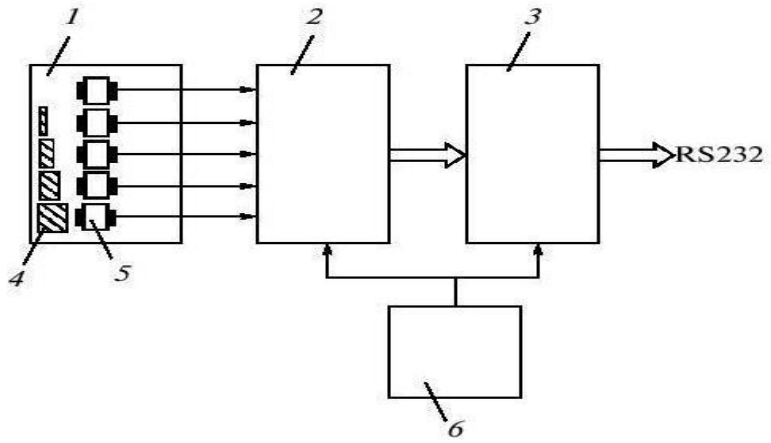


Рис. 1. Функциональная схема алмазного спектрометра (*1* – модуль фильтров и алмазных датчиков; *2* – модуль усиления, формирования и первичной обработки информации; *3* – модуль управления и информационного интерфейса; *4* – фильтры; *5* – чувствительные элементы на основе алмазной пластины толщиной 0,3 мм; *6* – модуль питания)

Однако информация о частицах ионизирующего излучения, содержащаяся в каждом счетчике, представляет собой суммарное значение потоков электронов, протонов и тяжелых заряженных частиц с различными энергиями. Для использования информации в системах управления потоками ионизирующего излучения необходимо по измерительным данным спектрометра получить раздельные значения потоков ионизирующего излучения по различным энергетическим диапазонам. Далее рассматриваются вопросы применения искусственной нейронной сети (ИНС) для обработки измерительных данных спектрометра потоков ионизирующего излучения на алмазных детекторах, что обеспечивает получение текущих значений плотностей потоков электронов, протонов и тяжелых заряженных частиц по 21 энергетическому диапазону. В результате, за время цикла измерения в 24 счетчиках частиц спектрометра фиксируются данные о количестве частиц соответствующего энергетического диапазона.

Важным этапом исследований является «обучение» нейронной сети, что в итоге позволяет ей с большой вероятностью правильно реагировать на новые, не предъявленные ей ранее данные. Для обучения нейронной сети необходимо сформировать базу данных, на основе которых может быть построена нейронная сеть прямого распространения, имеющая, как правило, б**о**льшую степень уверенности. В ряде случаев используется обучение с помощью алгоритма обратного распространения, являющегося модификацией градиентного спуска, при котором определяется среднеквадратичная ошибка сети как разница между эталонным и выходным вектором.

Обучение искусственной нейронной сети предполагается выполнить следующим образом:

1. Выполнить расчет с помощью алгоритма моделирования в программном пакете GEANT4 [2] данных спектрометра по экспериментально зарегистрированным спектрам ионизирующего излучения с использованием модели преобразования информации.

2. Осуществить обучение искусственной нейронной сети по экспериментально зарегистрированным спектрам потоков ионизирующего излучения и рассчитанным в предыдущем пункте данным спектрометра.

Среди большого разнообразия современных нейросетевых подходов предлагается решать задачу на базе персептрона с одним скрытым слоем. Выбор этого нейросетевого подхода обосновывается следствием теоремы Колмогорова [3], утверждающего, что «с помощью персептрона с одним скрытым слоем можно аппроксимировать любую сколь угодно сложную функцию с любой наперед заданной точностью». Структура персептрона с одним скрытым слоем представлена на рис. 2.

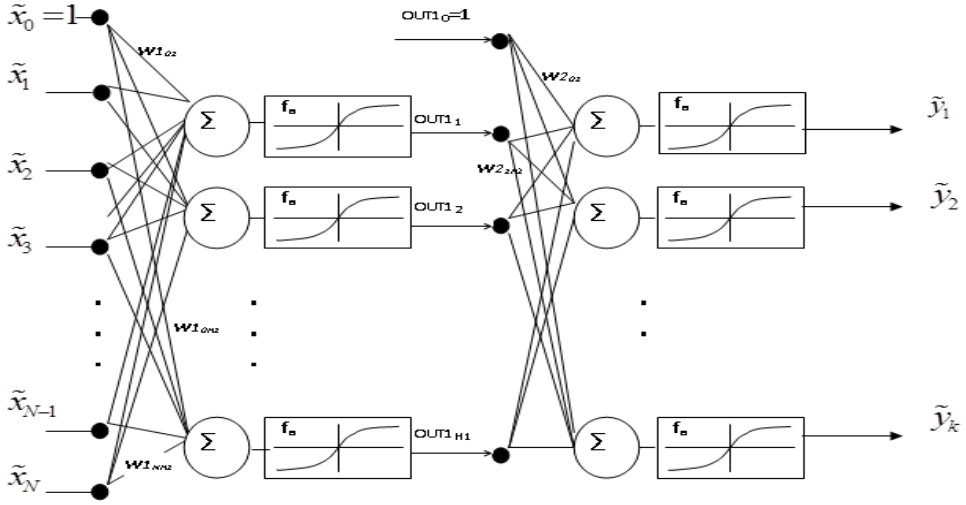


Рис. 2. Структура персептрона с одним скрытым слоем [4]

Использование нейросетевой технологии обеспечивает не только достаточно адекватное отображение модели исследуемого процесса или явления, возможность нахождения закономерностей в большом потоке противоречивой информации, сохранение связей между важными факторами, но и, что не менее важно, высокую скорость обработки данных и малую ресурсоемкость вычислительного оборудования. Дополнительным преимуществом этого подхода в настоящем исследовании является возможность решения представленной задачи в едином вычислительном базисе [3].

Выбор прототипа нейросетевой модели обосновывается двумя факторами. Первый - требованиями качества выхода, а именно относительная погрешность формируемых показаний спектрометра не должна превышать 10 %. Второй – простотой реализации, чем меньше слоев и нейронов в слоях, тем проще реализация, проще обучение и выше надежность. Так как определение прототипа нейросетевой модели относится к этапу экспериментальных исследований, то эта процедура, а также процедура обучения нейросетевой модели и проверки на адекватность функционирования нейросетевой модели проводится для каждого экспериментального образца нейросетевой модели. Для обучения нейросетевой модели необходимо определиться с алгоритмом минимизации функционала ошибки нейросетевой модели. В проведенных исследованиях использовался градиентный алгоритм наискорейшего спуска.

Для проведения экспериментальных исследований был разработан программный комплекс, позволяющий формировать и изменять конструктив нейросетевой модели, проводить ее обучение и тестирование, получать расчетные результаты и отображать их визуально и в виде таблиц. Интерфейс программного комплекса представлен на рис. 3–5.

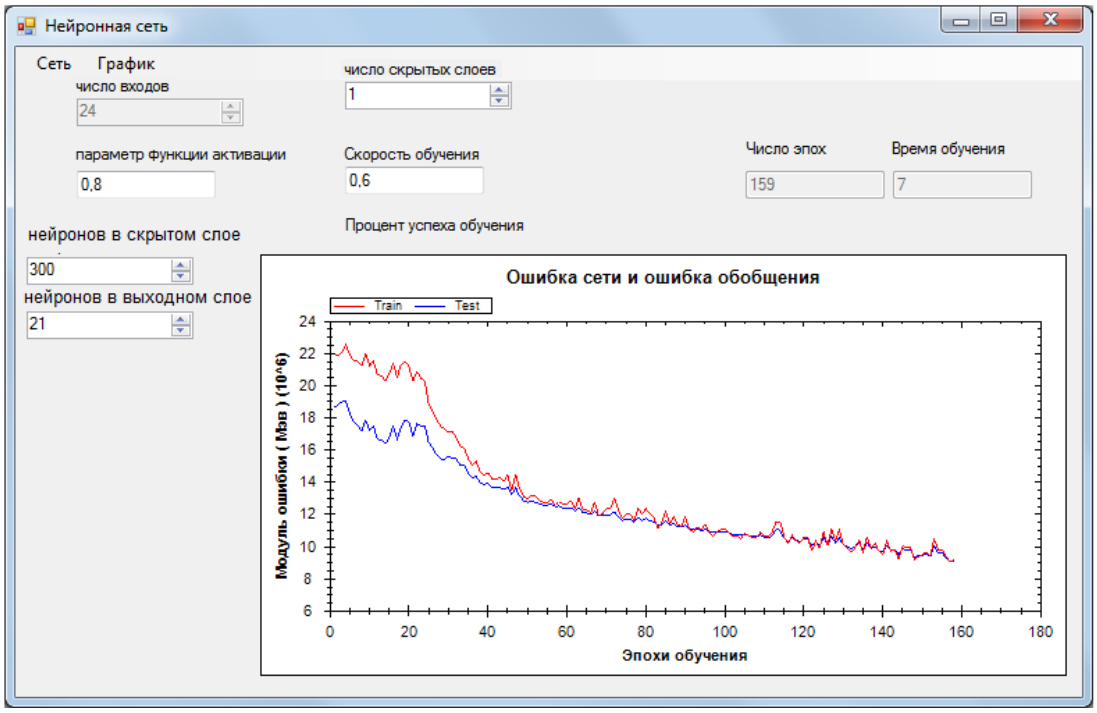


Рис. 3. Интерфейс для формирования структуры нейросетвой модели и ее обучения

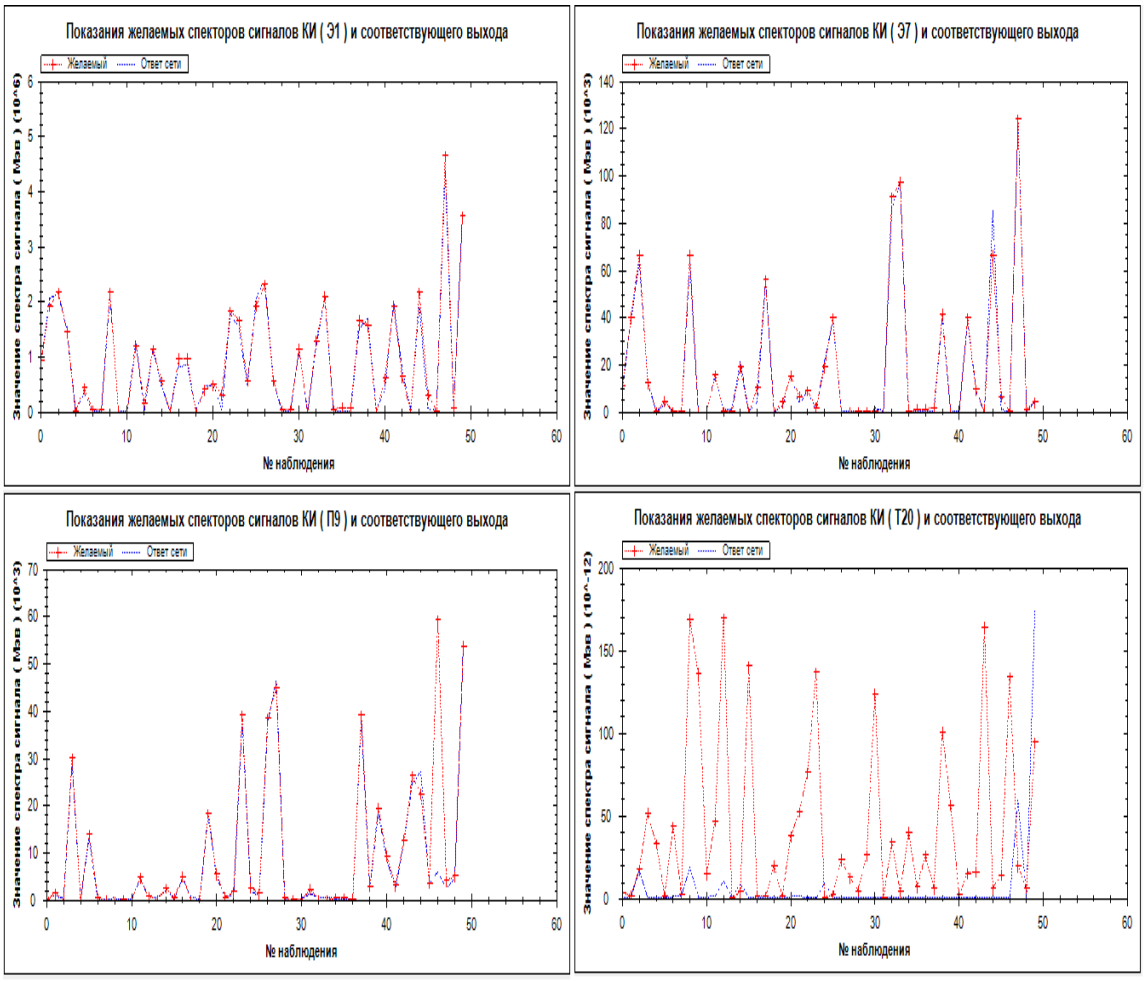


Рис. 4. Значения рассчитанных и полученных спектров сигналов ионизирующего излучения



Рис. 5. Гистограммы средних абсолютных ошибок по выходам нейронной сети [4]

Обучающая выборка измерительных данных потоков ионизирующего излучения делилась на две части: P = 800 примеров использовались для обучения нейронной сети (обучающие множество), 200 примеров – для тестирования (тестовое множество).

Примеры из обучающего множества равномерно перемешивались. Размерность входного сигнала N = 24, размерность выходного сигнала K = 21.

Первоначально количество нейронов в скрытом слое $H1$ выбиралось равным 80. Но такое количество нейронов не позволило обучить нейросетевую модель до значения ошибки $\varepsilon \leq 0{,}1$, т.е. с погрешностью менее 10 %. Только увеличив количество нейронов в скрытом слое $H1$ до 300, удалось добиться требуемой точности и то после 3 000 000 итераций обучения.

С использованием созданной нейросетевой модели было проведено моделирование преобразования информации о потоках ионизирующего излучения в тракте регистрации с измерением электронных потоков в диапазоне энергий от 0,1 до 5 МэВ и протонных потоков в диапазоне от 5 до 500 МэВ.

Полученные экспериментальные данные для всех пяти каналов регистрации потоков ионизирующего излучения спектрометра с учетом наличия ослабляющих фильтров в выбранных диапазонах энерговыделения позволили сформировать матрицу отклика спектрометра потокам ионизирующего излучения и использовать ее для восстановления данных о этих потоках.

В ходе выполнения данной работы были использованы интеллектуальные методы обработки информации в виде искусственной нейронной сети. Разработана нейросетевая модель и показаны возможности восстановления дифференцированных спектров электронного, протонного излучения и излучения тяжелых заряженных частиц по показаниям спектрометра на алмазных детекторах в энергетическом диапазоне от 0 до 10000 МэВ и реализовано программное обеспечение для ее обучения и тестирования. Модель преобразования информации в тракте регистрации спектрометра основывается на использовании алгоритма моделирования в программном пакете GEANT4.

Библиографический список

1. Spectrometric Diamond Detector of Fluxes of Ionizing Radiation for Space Transportation Systems / K. V. Zakharchenko, A. F. Kaperko, V. A. Kolyubin, V. P. Kulagin, S. A. L'vov, P. G. Nedosekin, E. N. Chumachenko // Measurement Techniques. – 2015. – Vol. 58, № 6. – P. 713–718.

2. Geant4: A toolkit for the simulation of the passage of particles through matter. – URL: http://geant4.web.cern.ch/geant4/ (дата обращения: 20.03.2017).

3. Колмогоров, А. Н. Элементы теории функций и функционального анализа / А. Н. Колмогоров, С. В. Фомин. – М. : Финансы и Статистика, 1976.

4. Применение нейросетевого подхода к измерениям потоков космического излучения / А. Ю. Истратов, К. В. Захарченко, А. Ф. Каперко, В. А. Колюбин, В. П. Кулагин, Р. И. Курочкин // Измерительная техника. – 2016. – № 3. – С. 49–54.

М. М. Алкезуини, В. И. Горбаченко

Пензенский государственный университет, Пенза, Россия

АДАПТАЦИЯ МЕТОДА ЛЕВЕНБЕРГА – МАРКВАРДТА ДЛЯ ОБУЧЕНИЯ СЕТЕЙ РАДИАЛЬНЫХ БАЗИСНЫХ ФУНКЦИЙ[1]

Разработана реализация метода Левенберга – Марквардта обучения сетей радиальных базисных функций. Даны рекомендации по выбору параметров обучения сетей. Создан комплекс программ в системе MATLAB, реализующий разработанные алгоритмы. Проведены экспериментальные исследования разработанных алгоритмов.

Сеть радиальных базисных функций (РБФ-сеть) описывается выражением

$$u(\mathbf{x}) = \sum_{k=1}^{n_{RBF}} w_k \varphi_k(\mathbf{x}), \tag{1}$$

где n_{RBF} – количество радиальных базисных функций (РБ-функций); w_k – вес k-й функции; $\mathbf{x}$ – входной вектор; $\varphi_k(\mathbf{x})$ – значение k-й РБ-функции.

В качестве РБ-функций применим функцию Гаусса (гауссиан) в двумерном случае имеющую вид

$$\varphi(\mathbf{x}) = \exp\left(-\frac{\|\mathbf{x}-\mathbf{c}\|^2}{2a^2}\right), \tag{2}$$

где $\mathbf{c} = [c_1, c_2]^{\mathrm{T}}$ – вектор координат центра РБ-функции, a – ширина (параметр формы), $\|\mathbf{x}-\mathbf{c}\| = \sqrt{(x_1-c_1)^2-(x_2-c_2)^2}$ – евклидова норма (расстояние между точкой $\mathbf{x}$ и центром $\mathbf{c}$ РБ-функции).

РБФ-сети с успехом используются для аппроксимации функций [2]. Для примера рассмотрим аппроксимацию функций двух переменных. В этом случае сеть имеет два входа – координаты точки $\mathbf{x} = [x_1, x_2]^{\mathrm{T}}$ и один выход – значение функции в точке. После обучения сеть при подаче на ее вход координат произвольной точки области определения функции выдает значение функции в этой точке.

Обучение РБФ-сети – минимизация функционала ошибки

$$I = \frac{1}{2}\sum_{j=1}^{n} e_j^2 = \frac{1}{2}\sum_{j=1}^{n}\left(u(\mathbf{p}_j - T_j)\right)^2, \tag{3}$$

где n – количество пробных точек; e_j – ошибка решения в j-й пробной точке; $\mathbf{p}_j$ – координаты j-й пробной точки (в случае аппроксимации функции двух переменных

[1] Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 16-08-00906.

$\mathbf{p}_j = \left[p_{j1}, p_{j2} \right]^{\mathrm{T}}$); $u\left(\mathbf{p}_j\right)$ – решение (1) в j-й пробной точке; T_j – целевое значение в j-й пробной точке, множитель 1/2 введен для упрощения вычислений.

Рассмотрим применение для обучения сети метода второго порядка – метода Левенберга-Марквардта [1], являющегося реализацией известного метода безусловной оптимизации [3]. Метод Левенберга-Марквардта широко применяется для обучения многослойных персептронов [1], но практически не применяется для обучения сетей радиальных базисных функций. Следует отметить работу [4], в которой предложена эффективная матрицы Гессе, рассмотрена реализация метода Гаусса-Ньютона для обучения РБФ-сетей, но не рассмотрена реализация метода Левенберга-Марквардта.

Рассмотрим настройку параметров. Введем единый вектор параметров

$$\boldsymbol{\theta} = \left[w_1, w_2, \ldots, w_{n_{RBF}}, c_{11}, c_{21}, \ldots, c_{n_{RBF}1}, c_{12}, c_{22}, \ldots, c_{n_{RBF}2}, a_1, a_2, \ldots, a_{n_{RBF}} \right]^{\mathrm{T}},$$

где параметры j-й РБ-функции ($j = 1, 2, 3, \ldots, n_{RBF}$): w_j – вес, c_{j1} и c_{j2} – координаты центра (рассматриваем аппроксимацию функций двух переменных), a_j – ширина.

Настройка вектора параметров $\boldsymbol{\theta}$ в k-м цикле (итерации) производится по формуле $\boldsymbol{\theta}^{(k)} = \boldsymbol{\theta}^{(k-1)} + \Delta\boldsymbol{\theta}^{(k)}$, где вектор поправки $\Delta\boldsymbol{\theta}^{(k)}$ находится из решения системы линейных алгебраических уравнений

$$\left(\mathbf{J}_{k-1}^{\mathrm{T}} \mathbf{J}_{k-1} + \mu_k \mathbf{E} \right) \Delta\boldsymbol{\theta}^{(k)} = -\mathbf{g}_{k-1} \tag{4}$$

где $\mathbf{E}$ – единичная матрица; μ_k – параметр регуляризации, изменяющийся на каждом шаге обучения; $\mathbf{g} = \mathbf{J}^{\mathrm{T}}\mathbf{e}$ – вектор градиента функционала (3) по вектору параметров $\boldsymbol{\theta}$; $\mathbf{e} = \left[e_1 \; e_2 \ldots e_n \right]^{\mathrm{T}}$ – вектор ошибок; $\mathbf{J}_{k-1}$ – матрица Якоби, вычисленная по значениям параметров сети в $k-1$ итерации.

Матрица Якоби имеет вид

$$\mathbf{J} = \left[\begin{array}{ccc:ccc:ccc:ccc} \frac{\partial e_1}{\partial w_1} & \cdots & \frac{\partial e_1}{\partial w_{n_{RBF}}} & \frac{\partial e_1}{\partial c_{11}} & \cdots & \frac{\partial e_1}{\partial c_{n_{RBF}1}} & \frac{\partial e_1}{\partial c_{12}} & \cdots & \frac{\partial e_1}{\partial c_{n_{RBF}2}} & \frac{\partial e_1}{\partial a_1} & \cdots & \frac{\partial e_1}{\partial a_{n_{RBF}}} \\ \frac{\partial e_2}{\partial w_1} & \cdots & \frac{\partial e_2}{\partial w_{n_{RBF}}} & \frac{\partial e_2}{\partial c_{11}} & \cdots & \frac{\partial e_2}{\partial c_{n_{RBF}1}} & \frac{\partial e_2}{\partial c_{12}} & \cdots & \frac{\partial e_2}{\partial c_{n_{RBF}2}} & \frac{\partial e_2}{\partial a_1} & \cdots & \frac{\partial e_2}{\partial a_{n_{RBF}}} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \frac{\partial e_n}{\partial w_1} & \cdots & \frac{\partial e_n}{\partial w_{n_{RBF}}} & \frac{\partial e_n}{\partial c_{11}} & \cdots & \frac{\partial e_n}{\partial c_{n_{RBF}1}} & \frac{\partial e_n}{\partial c_{12}} & \cdots & \frac{\partial e_n}{\partial c_{n_{RBF}2}} & \frac{\partial e_n}{\partial a_1} & \cdots & \frac{\partial e_n}{\partial a_{n_{RBF}}} \end{array} \right].$$

Представим матрицу Якоби в блочном виде

$$\mathbf{J} = \left[\mathbf{J}_{\mathbf{w}} \,\vdots\, \mathbf{J}_{\mathbf{c}_1} \,\vdots\, \mathbf{J}_{\mathbf{c}_2} \,\vdots\, \mathbf{J}_{\mathbf{a}} \right],$$

где

$$\mathbf{J}_{\mathbf{w}} = \left[\begin{array}{cccc} \frac{\partial e_1}{\partial w_1} & \frac{\partial e_1}{\partial w_2} & \cdots & \frac{\partial e_1}{\partial w_{n_{RBF}}} \\ \frac{\partial e_2}{\partial w_1} & \frac{\partial e_2}{\partial w_2} & \cdots & \frac{\partial e_2}{\partial w_{n_{RBF}}} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial e_n}{\partial w_1} & \frac{\partial e_n}{\partial w_2} & \cdots & \frac{\partial e_n}{\partial w_{n_{RBF}}} \end{array} \right], \quad \mathbf{J}_{c_1} = \left[\begin{array}{ccc} \frac{\partial e_1}{\partial c_{11}} & \cdots & \frac{\partial e_1}{\partial c_{n_{RBF}1}} \\ \frac{\partial e_2}{\partial c_{11}} & \cdots & \frac{\partial e_2}{\partial c_{n_{RBF}1}} \\ \cdots & \cdots & \cdots \\ \frac{\partial e_n}{\partial c_{11}} & \cdots & \frac{\partial e_n}{\partial c_{n_{RBF}1}} \end{array} \right], \quad \mathbf{J}_{c_2} = \left[\begin{array}{ccc} \frac{\partial e_1}{\partial c_{12}} & \cdots & \frac{\partial e_1}{\partial c_{n_{RBF}2}} \\ \frac{\partial e_2}{\partial c_{12}} & \cdots & \frac{\partial e_2}{\partial c_{n_{RBF}2}} \\ \cdots & \cdots & \cdots \\ \frac{\partial e_n}{\partial c_{12}} & \cdots & \frac{\partial e_n}{\partial c_{n_{RBF}2}} \end{array} \right]$$

$$\mathbf{J}_{c_1}=\begin{bmatrix}\frac{\partial e_1}{\partial c_{11}} & \cdots & \frac{\partial e_1}{\partial c_{n_{RBF}1}}\\ \frac{\partial e_2}{\partial c_{11}} & \cdots & \frac{\partial e_2}{\partial c_{n_{RBF}1}}\\ \cdots & \cdots & \cdots\\ \frac{\partial e_n}{\partial c_{11}} & \cdots & \frac{\partial e_n}{\partial c_{n_{RBF}1}}\end{bmatrix},\quad \mathbf{J}_{c_2}=\begin{bmatrix}\frac{\partial e_1}{\partial c_{12}} & \cdots & \frac{\partial e_1}{\partial c_{n_{RBF}2}}\\ \frac{\partial e_2}{\partial c_{12}} & \cdots & \frac{\partial e_2}{\partial c_{n_{RBF}2}}\\ \cdots & \cdots & \cdots\\ \frac{\partial e_n}{\partial c_{12}} & \cdots & \frac{\partial e_n}{\partial c_{n_{RBF}2}}\end{bmatrix},\quad \mathbf{J}_{a}=\begin{bmatrix}\frac{\partial e_1}{\partial a_1} & \cdots & \frac{\partial e_1}{\partial a_{n_{RBF}}}\\ \frac{\partial e_2}{\partial a_1} & \cdots & \frac{\partial e_2}{\partial a_{n_{RBF}}}\\ \cdots & \cdots & \cdots\\ \frac{\partial e_n}{\partial a_1} & \cdots & \frac{\partial e_n}{\partial a_{n_{RBF}}}\end{bmatrix}.$$

Элементы матрицы $\mathbf{J}_{\mathbf{w}}$ с учетом (3) и (1) имеют вид

$$\frac{\partial e_i}{\partial w_j}=\frac{\partial}{\partial w_j}\left[u(\mathbf{p}_i)-T_i\right]=\varphi_j(\mathbf{p}_i), \tag{5}$$

где $\varphi_j(\mathbf{p}_i)$ – значение j-й радиальной РБ-функции (2) в пробной точке $\mathbf{p}_i$.

Элементы матрицы $\mathbf{J}_{c_1}$ описываются формулой

$$\frac{\partial e_i}{\partial c_{j1}}=\frac{\partial}{\partial c_{j1}}\left[u(\mathbf{p}_i)-T_i\right]=w_j\varphi_j(\mathbf{p}_i)\cdot\frac{p_{i1}-c_{j1}}{a_j^2}.$$

Аналогично для элементов матрицы $\mathbf{J}_{c_2}$ получаем

$$\frac{\partial e_i}{\partial c_{j2}}=w_j\cdot\varphi_j(\mathbf{p}_i)\cdot\frac{p_{i2}-c_{j2}}{a_j^2}.$$

Элементы матрицы $\mathbf{J}_a$ вычисляются по формуле

$$\frac{\partial e_i}{\partial a_j}=\frac{\partial}{\partial a_j}\left[u(\mathbf{p}_i)-T_i\right]=w_j\varphi_j(\mathbf{p}_i)\cdot\frac{\left\|\mathbf{p}_i-\mathbf{c}_j\right\|^2}{a_j^3}.$$

Матрица $\mathbf{J}_{k-1}^{\mathrm{T}}\mathbf{J}_{k-1}+\mu_k\mathbf{E}$ является плотной симметричной и положительно определенной. Поэтому для решения системы (4) целесообразно использовать метод Холецкого [5]. Недостатком метода Холецкого является использование при выполнении разложения матриц длительной операции извлечения квадратного корня. От этого недостатка свободен метод $\mathrm{LDL}^{\mathrm{T}}$ – разложения [5], который представляет матрицу в виде $\mathbf{A}=\mathbf{LDL}^{\mathrm{T}}$, где $\mathbf{L}$ – нижняя треугольная матрица с единичной главной диагональю, $\mathbf{D}$ – диагональная матрица, T – операция транспонирования матрицы.

В методе Левенберга-Марквардта важен правильный подбор параметра μ. В начале процесса обучения, когда вектор весов далек от оптимального, используется относительно большое значение параметра μ. В этом случае Гессиан заменяется приближенным значением $\mathbf{H}\approx\mu\mathbf{E}$, а вектор поправки определяется методом градиентного спуска с малым шагом: $\Delta\boldsymbol{\theta}^{(k)}=-1/\mu_k\cdot\mathbf{g}_{k-1}$. По мере уменьшения погрешности параметр μ уменьшается и метод приближается к методу Ньютона с аппроксимацией Гессиана $\mathbf{H}\approx\mathbf{J}^{\mathrm{T}}\mathbf{J}$. Это обеспечивает высокую скорость сходимости, так как метод Ньютона вблизи минимума функционала ошибки имеет хорошую сходимость. Д. Марквардт рекомендуется [6] начинать со значения μ_0 и коэффициента $\nu>1$. Текущее значение μ делится на ν, если функционал ошибки уменьшается, или умножается на ν, если функционал ошибки увеличивается. Процесс заканчивается при малом значении функционала ошибки (3) или среднеквадратической погрешности

$$I_{MSE}=\frac{1}{n}\sum_{j=1}^{n}\left(u(\mathbf{p}_j)-T_j\right)^2. \tag{6}$$

Можно показать, что метод Левенберга-Марквардта является вариантом метода доверительных областей (trust-region methods) [7]. Особенностью метода доверительных областей является поиск минимума функции в ограниченной доверительной области. В [8] разработана реализация метода доверительных областей для обучения сетей радиальных базисных функций. Но метод доверительных областей является достаточно сложным, так как из-за условия поиска минимума в доверительной области требует решения задачи условной оптимизации, в отличие от других градиентных методов обучения нейронных сетей, в которых решается задача безусловной оптимизации. Марквардт доказал [6], что метод Левенберга-Марквардта эквивалентен методу доверительных областей, при этом радиус доверительной области регулируется параметром μ. Таким образом, метод Левенберга-Марквардта сохраняя положительные качества метода доверительных областей, является более простым в реализации методом.

Экспериментальное исследование разработанного алгоритма проводилось на примере аппроксимации функции $z = x^2 + y^2$ в области $(x = -3 \ldots +3,\ y = -3 \ldots +3)$. Количество узлов интерполяции равно 100. Узлы интерполяции располагались случайным образом в области. Количество РБ-функций равно 16. В начальном состоянии центры РБ-функций располагались в области на сетке с шагом, равным единице. Веса инициировались случайными числами, равномерно распределенными от 0 до 0,001. Начальная ширина всех РБ-функций была постоянной, равной 1,0. Итерационный процесс обучения продолжался до достижения значения средней квадратической ошибки (6), равной 0,01. Для проведения экспериментов разработан комплекс программ в системе MATLAB. Эксперименты проводились на компьютере со следующими характеристиками: процессор Intel Core i5-2500K, 3,30 GHz, ОЗУ 8,0 GB. Результаты экспериментов представлены в таблице. Так как число итераций и время решения зависят от случайных начальных значений весов, то для каждого метода проводилось 10 экспериментов и в таблице представлены полученные диапазоны числа итераций и времени решения. Значения коэффициентов подбирались экспериментально. В таблице приведены также результаты для метода градиентного спуска, применяемого для обучения РБФ-сетей. Для этого метода индексы у коэффициентов означают: 1 – коэффициент для весов, 2 – для центров, 3 – для ширины.

Таблица

Результаты экспериментов

Метод	Параметры	Число итераций	Время решения, С
Градиентный спуск	$\eta_1 = 0,00150$, $\eta_2 = 0,00100$, $\eta_3 = 0,00050$	45000–70000	2500–5000
Метод Левенберга-Марквардта	$\mu_0 = 0,1, \quad \nu = 10$	6–11	1,77–1,96

Метод Левенберга-Марквардта обеспечивает качественное преимущество по сравнению с методом градиентного спуска. С помощью метода Левенберга-Марквардта удалось достичь значения среднеквадратической погрешности, равного 10^{-6}, что практически невозможно сделать методами первого порядка. Решение при этом было достигнуто за число циклов обучения, изменявшееся от 23 до 98 (время решения менялось от 2,23 С до 4,65 С).

Недостаток метода Левенберга-Марквардта – плохая обусловленность системы (4), зависящая от начальных значений ширины РБ-функций и увеличивающаяся с ростом точности вычислений. Известно [9], что матрица, элементами которой являются РБ-функции, является плохо обусловленной и обусловленность матрицы зависит от ширины РБ-функций. С ростом ширины значения РБ-функций (2), являющиеся элементами матрицы $\mathbf{J}_\mathbf{w}$ (5), стремятся к единице, а элементы матриц $\mathbf{J}_\mathbf{c}$ и $\mathbf{J}_a$ стремятся к нулю. Число обусловленности матрицы $\mathbf{J}^\mathrm{T}\mathbf{J}$ растет. Параметр регуляризации μ улучшает обусловленность системы (4), но уменьшение параметра μ по мере уменьшения погрешности приводит к ухудшению обусловленности. В отличие от обучения многослойного персептрона методом Левенберга-Марквардта, обучение РБФ-сети требует значительно больших значений

параметра регуляризации μ. Так для обучения многослойного персептрона рекомендуется [10] $\mu = 0{,}001$, а обучение РБФ-сети работает даже при $\mu > 1$, но при этом процесс изменения среднеквадратической погрешности носит сильно колебательный характер.

Библиографический список

1. Осовский, С. Нейронные сети для обработки информации / С. Осовский. – М. : Горячая линия-Телеком, 2016. – 448 с.
2. Buhmann, M. D. Radial Basis Functions: Theory and Implementations / M. D. Buhmann. – Cambridge University Press, 2009. – 272 p.
3. Гилл, Ф. Практическая оптимизация / Ф. Гилл, У. Мюррей, М. Райт. – М. : Мир, 1985. – 509 с.
4. Fast and Efficient Second-Order Method for Training Radial Basis Function Networks / T. Xie, H. Yu, J. Hewlett, P. Rozycki, B. Wilamowski // IEEE transactions on neural networks and learning systems. – 2012. – Vol. 23, № 4. – P. 609–619.
5. Горбаченко, В. И. Вычислительная линейная алгебра с примерами на MATLAB / В. И. Горбаченко. – СПб. : БХВ-Петербург, 2011. – 320 с.
6. Marquardt, D. W. An Algorithm for Least-Squares Estimation of Nonlinear Parameters / D. W. Marquardt // Journal of the Society for Industrial and Applied Mathematics. – 1963. – Vol. 11, № 2. – P. 431–441.
7. Conn, A. R. Trust regions methods / A. R. Conn, N. I. M. Gould, P. L. Toint. – Society for Industrial and Applied Mathematics, 2000. – 959 p.
8. Горбаченко В. И. Решение краевых задач математической физики с помощью сетей радиальных базисных функций / В. И. Горбаченко, М. В. Жуков // Журнал вычислительной математики и математической физики. – 2017. – Т. 57, № 1. – С. 133–143.
9. Boyd, J. P. Numerical experiments on the condition number of the interpolation matrices for radial basis functions / J. P. Boyd, K. W. Gildersleeve // Applied Numerical Mathematics. – 2011. – Vol. 61, Issue 4. – P. 443–459.
10. Beale, M. H. Neural Network Toolbox. User's Guide / M. H. Beale, M. T. Hagan, H. B. Demuth. – Natick : MathWorks, Inc., 2017. – 446 p.

М. Ю. Бабич

АО «НПП «Рубин», Пенза, Россия

ОПРЕДЕЛЕНИЕ СТЕПЕНИ ВОЗМОЖНОЙ ИНТЕЛЛЕКТУАЛИЗАЦИИ ОРГАНИЗАЦИОННО-ТЕХНИЧЕСКИХ СИСТЕМ ПОСРЕДСТВОМ ФУНКЦИИ ПРИНАДЛЕЖНОСТИ ЛИЦА, ПРИНИМАЮЩЕГО РЕШЕНИЯ, К АВТОМАТИЗИРОВАННОЙ СИСТЕМЕ УПРАВЛЕНИЯ

Анализируется возможность интеллектуализации задач для лиц, принимающих решение, в многоагентной организационно-технической системе. Приводится метод определения степени интеллектуализации, основанный на принадлежности агентов виртуальной и реальной информационным системам.

Рассмотрим сложные организационно-технические системы в качестве многоагентных систем. Агентами в организационно-технических системах рассматриваемого класса являются люди или люди, управляющие техническими устройствами. Информация, циркулирующая в подобных системах, характеризуется большим объемом, своей неполнотой, наличием в ней нечетких, противоречивых данных. Информационные процессы, протекающие в системах, достаточно сложные и не являются независимыми. Важной подсистемой является подсистема управления, реализующая мягкое управляющее воздействие на объекты управления, находящиеся в контуре системы. Время принятия решения близко к реаль-

ному. К классу подобных многоагентных систем можно отнести достаточно большое число систем: системы управления транспортом, системы МЧС, системы управления силовыми операциями и другие. Более подробные свойства многоагентных систем рассматриваемого класса приведены в [1, 2].

Под интеллектуализацией решения задач принятия решений в подсистеме управления будем понимать процесс формализации задач и реализации их решения на базе современного математического аппарата средствами вычислительной техники. В [2] были приведены причины, затрудняющие внедрение средств интеллектуализации в многоагентных системах рассматриваемого класса. Многие задачи после внедрения не реализуются в процессе принятия решения должностными лицами в подсистеме управления. Поэтому в процессе проектирования и создания систем возникает необходимость в определении степени интеллектуализации подсистемы управления многоагентных организационно-технических систем рассматриваемого класса, которая позволила бы более эффективно использовать решение задач интеллектуальной поддержки управления.

Методика расчета степени интеллектуализации основана на предположении о справедливости аксиом принадлежности интеллектуального агента более чем одной системе [3], понятии нечеткой точки плоскости [4] и алгоритме поведения интеллектуального агента на основе изменения ментального типа системы, которой принадлежит агент [5].

Обозначим систему, в которой функционирует агент, через S. Из системы S выделим множество агентов A, состоящее из рациональных агентов a. Через P_a и P_S обозначим цели агента и системы. Если агент принадлежит системе, то есть $a \in A \subset S$, то будем считать, что агент функционирует так, чтобы была достигнута цель системы или ее часть, то есть цель P_a достигается в интересах цели P_S. Введем понятие суперсистемы W, которое ограничивает множество систем сверху. Суперсистем может быть несколько, системы S могут принадлежать только одной из суперсистем, но система, состоящая из нескольких суперсистем, невозможна.

Аксиома принадлежности интеллектуального агента системам утверждает, что любой агент принадлежит одновременно нескольким разным системам, каждая из которых не является подмножеством другой. Цели систем, которым принадлежит агент, не совпадают:

$$\forall a \in A, \exists S_1, S_2, \text{ что } (a \in S_1) \wedge (a \in S_2) \wedge (S_1 \neq S_2) \wedge (S_1 \not\subset S_2) \wedge (S_2 \not\subset S_1) \wedge$$
$$\wedge (S_1 \subseteq W_1) \wedge ((S_2 \subseteq W_1) \vee (S_2 \subseteq W_2)) \wedge (P_{S1} \neq P_{S2}). \quad (1)$$

Заметим, что должностное лицо, принимающее решение (ЛПР), принадлежит двум системам S_1, S_2, где S_1 – реальная система, S_2 – виртуальная.

Под реальной системой будем понимать систему, обрабатывающую реальную информацию, то есть информацию о состоянии системы, внешней среды, множестве агентов. Реальная информация является объективной и не зависит от таких факторов, как неточность передачи данных, ошибки в программном обеспечении и в алгоритмах обработки информации, ошибки, вызванные неточностью средств наблюдения, используемой БД и т.д. Суперсистеме необходимо, чтобы ЛПР принимало решения, являющиеся реально эффективными. Она не учитывает, каким образом приняты эти решения: с учетом решения задач интеллектуальной поддержки или без них.

Под виртуальной системой будем понимать систему, обрабатывающую виртуальную информацию, то есть информацию, в поле которой работают ЛПР. В виртуальной информации все приведенные факторы возможного ее искажения могут присутствовать. В подсистеме управления информацию обрабатывают средства вычислительной техники и программное обеспечение. ЛПР вынуждены принимать решения на основе виртуальной информации.

Реальная система принадлежит суперсистеме, которая требует, чтобы работа ЛПР была эффективна в реальных ситуациях. Цели, которые пытается достичь ЛПР в виртуальной системе, ставятся на основе информации, обработанной средствами вычислительной техники, на основе решения задач интеллектуальной поддержки. Таким образом, цели реальной и виртуальной систем могут не совпадать.

Рассмотрим объекты нечеткой геометрии.

Предположим, что плоскости Z принадлежит некоторая окрестность U обычной, четкой точки. Введем характеристическую функцию принадлежности $\mu(z)$ с множеством принадлежности [0, 1]. Пусть $U^{н}$ – это нечеткое множество универсального множества Z, соответствующее функции принадлежности $\mu(z)$. Так как каждая точка на плоскости определяется парой координат (x, y), то $\mu(z) = \mu(x, y)$. Функция принадлежности $\mu(x, y)$ определяет некоторое нечеткое бинарное отношение:

$$R: (X, Y) \rightarrow [0, 1], \quad (2)$$

где X, Y – множества значений координат x, y.

Введем следующие условия.

У1. Для любой точки (x, y), принадлежащей U, выполняется: $0 < \mu(x, y) \leq 1$.

У2. Для любой точки (x, y), не принадлежащей U, выполняется: $\mu(x, y) = 0$.

У3. Существует одна единственная точка (x_0, y_0), принадлежащая U, в которой выполняется: $\mu(x_0, y_0) = \max\mu(x, y)$, где (x, y) – любая точка, принадлежащая Z.

Значение функции принадлежности $\mu(x, y)$ соответствует степени уверенности, что объект, местоположение которого определяется нечетким точечным объектом, имеет координаты (x, y).

В дальнейшем от функции принадлежности $\mu(x, y)$ можно перейти к функции принадлежности $\mu(r)$, где r – обычное расстояние от четкой точки (ядра) окрестности U.

Степень интеллектуализации можно понимать как принадлежность нечеткой точке, ядро которой соответствует максимальной степени интеллектуализации.

Будем предполагать, что принадлежность агента a некоторой системе S определяется правилами поведения агента $\varphi(a)$, то есть выполнением некоторых регламентированных действий по отношению к системе. Каждая система S имеет свой тип и свою степень требовательности к выполнениям правил. Обозначим через $type(S)$ степень необходимости выполнения правила φ в соответствии с типом системы.

Каждый агент определяет для себя тип системы S, значение $type(S)$ и вносит их в свои ментальные понятия, убеждения. Обозначим степень необходимости выполнения правила $\varphi(a)$, которое внесено в ментальные понятия агента, через $type^S(a)$. В общем случае значение $type^S(a)$ может быть отличным от значения $type(S)$, но $type(S) \geq type^S(a)$. Агент действует не в соответствии со значением $type(S)$, а исходя из своих ментальных убеждений, исходя из значения $type^S(a)$.

На множествах значений $type(S)$ и $type^S(a)$ введем отношение линейного квазипорядка, дающее возможность сравнивать значения. Чем больше значение $type(S)$, тем строже относится система к агентам, ей принадлежащим.

Значение $type^S(a)$ влияет на поведенческие свойства агента в системе S. При отсутствии допустимых алгоритмов достижения двух поставленных целей одновременно агент вынужден отказаться от выполнения алгоритма достижения какой-либо одной или обеих целей, предварительно изменив свои ментальные понятия. Изменение ментального значения $type^S(a)$ в процессе функционирования системы и агента может зависеть от различных параметров и происходить с различной скоростью. Если представить значение этих параметров через z, то ментальное значение степени необходимости выполнения правила $\varphi(a)$ у агента является функцией от переменной z, т.е. $type^S(a) = type^S(a)\,(z)$. Чем меньше разность $type(S) - type^S(a)\,(z)$, тем больше действительная степень принадлежности агента a к системе S.

Степень принадлежности агента a к системе S представима в качестве принадлежности нечеткой точке с функцией принадлежности $\mu(z)$, обладающей свойствами

$$\mu(z) = 0, \text{ если } type^S(a)\,(z) = 0,$$
$$\mu(z) = 1, \text{ если } type(S) = type^S(a)\,(z). \quad (3)$$

Например, $\mu(z) = type^S(a)\,(z) / type(S)$.

Можно предположить, что S – это виртуальная система, $type(S)$ – степень идеальной интеллектуализации виртуальной системы, которая равна 1, $type^S(a)$ – ментальное отражение $type(S)$ у ЛПР, а z – параметр, от которого зависит интеллектуализация системы. Тогда $\mu(z)$ – функция принадлежности ЛПР к виртуальной системе или возможная степень интеллектуализации, зависящая от параметра z. Если таких параметров несколько, т.е. $z = z_1 \wedge z_2 \wedge \ldots \wedge z_m$.

Приведем примеры параметров z_i:

- частота необходимости решения задач со средствами интеллектуальной поддержки;
- частота адекватности решения задач со средствами интеллектуальной поддержки или обратное – частота неадекватности их решения;
- степень повторяемости событий для задачи, то есть степень повторяемости состояний системы и внешней среды;
- степень возможности наличия неучтенных (неожиданных для ЛПР и задач) параметров;
- степень наличия когнитивных процессов, в которых ЛПР имеет преимущество при решении задач по сравнению с задачами, решаемыми на средствах вычислительной техники;
- степень участия ЛПР в разработке других систем интеллектуальной поддержки.

Библиографический список

1. Бабич, М. Ю. Условия отсутствия адекватных решений некоторых задач интеллектуальной поддержки управления в процессе функционирования многоагентных систем / М. Ю. Бабич // Проблемы теоретической кибернетики : тр. XVIII Междунар. конф. (Пенза, 19–23 июня 2017 г.). – М. : МАКС Пресс, 2017. – С. 39–42

2. Бабич, М. Ю. Решение задач системы поддержки принятия решений в процессе управления распределенными, динамическими ресурсами / М. Ю. Бабич // Информационно-измерительные и управляющие системы. – 2014. – Т. 12, № 4. – С. 12–18.

3. Бабич, М. Ю. Алгоритмы достижения целей адаптивных агентов в контуре нескольких организационно-технических систем / М. Ю. Бабич, А. М. Бабич // XXI век: итоги прошлого и проблемы настоящего плюс. Сер.: Технические науки. Информационные технологии. – 2015. – Вып. 3 (25). – С. 77–81.

4. Бабич, М. Ю. Вопросы пользовательской нечеткой картографии // Вопросы радиоэлектроники. Сер.: ЭВТ. – 2007. – Вып. 1. – С. 141–148.

5. Бабич, М. Ю.Общий алгоритм функционирования систем и агентов в случае выполнения условий аксиом принадлежности // XXI век: итоги прошлого и проблемы настоящего плюс. Сер.: Технические науки. Информатика, вычислительная техника и управление. – 2016. – Вып. 3 (31). – С. 85–89.

Хоанг Тхай Хо

Пензенский государственный университет, Пенза, Россия

КОМПЛЕКС ПРОГРАММ ГЕОМЕТРИЧЕСКОГО МОДЕЛИРОВАНИЯ ПРОСТРАНСТВЕННЫХ ФОРМ НА ОСНОВЕ ЧИСЛЕННОЙ РЕАЛИЗАЦИИ ИНТЕРПОЛЯЦИОННЫХ АЛГОРИТМОВ

Описаны математический аппарат и программная реализация компьютерного моделирования пространственных форм в соответствии с требованиями визуализации. Применяется двухэтапная интерполяция. На первом этапе используются радиальные базисные функции. Результатом являются новые опорные точки, принадлежащие поверхности и равномерно расставленные в пространстве. На втором этапе используется B-сплайновая интерполяция с точным прохождением поверхности через новые опорные точки.

Геометрическое моделирование пространственных форм с целью их дальнейшей визуализации имеет множество приложений. Это научная визуализация, мониторинг динамических объектов, геоинформационные системы, проектирование объектов строительства и архитектуры, транспортные симуляторы и другое.

Объекты визуализации обычно задаются пространственным расположением опорных точек, принадлежащих поверхности этих объектов. Реконструкция поверхности по опорным точкам осуществляется, как правило, интерполяционными методами. В процессе реконструкции объектов визуализации нужно учитывать следующие основные требования:

- точное прохождение поверхности через характерные точки;
- сохранение топологической тенденции в промежутках между опорными точками;
- гладкость поверхности и отсутствие у нее осцилляций;
- возможность применить быстрые алгоритмы для вычисления промежуточных точек поверхности;
- возможность алгоритмически просто получить полигональную модель.

Следует также отметить, что неравномерная расстановка опорных точек в пространстве затрудняет применение быстрых сеточных методов моделирования и полигонизации поверхности. Возникает задача выполнения геометрического моделирования и визуализации пространственных форм в соответствии с перечисленными требованиями.

Можно предложить двухэтапную организацию геометрического моделирования пространственных форм на основе численной реализации. Для исследования изобразительных возможностей интерполяционных методов моделирования и рационального выбора характеристик модели разработан комплекс из двух программ. Они реализуют два этапа моделирования.

На первом этапе осуществляется переход от исходных опорных точек к новым опорным точкам, равномерно расставленным в пространстве. Для этого используется интерполяция радиальными базисными функциями (РБФ). Новыми опорными точками становятся точки полученной поверхности, вычисленные с заданным шагом.

Для получения поверхности ядра РБФ размещаются в опорных точках. Координата текущей точки поверхности находится как сумма взвешенных значений РБФ. Степень влияния опорной точки на текущую точку («вес» опорной точки) зависит от расстояния r между ними, а также от вида РБФ, выбранной для опорной точки.

В случае описания поверхности однозначными функциями для упрощения вычислений можно уменьшить размерность пространства для нахождения r до двух, то есть вычислять не расстояние между самими точками в трехмерном пространстве, а расстояние между проекциями этих точек на поверхность аргументов [1]. В этом случае область влияния каждой опорной точки представляет собой круг. Используя предположение об однозначности интерполянта, поверхность можно описать в явной форме:

$$z(x,y)=\sum_{i=1}^{N}\lambda_i\varphi(r_i),\qquad r_i=\sqrt{(x-x_i)^2+(y-y_i)^2}\ , \tag{1}$$

где x, y, z – координаты текущей точки поверхности;

r_i – расстояние между i-й опорной точкой и текущей точкой на плоскости xy;

$\varphi(r_i)$ – значение радиальной базисной функции i-й опорной точки;

λ_i – вес i-й опорной точки;

x_i, y_i – координаты i-й опорной точки на плоскости xy;

N – количество опорных точек.

Весовые коэффициенты λ_i находятся из системы линейных уравнений, полученных из (1) для случаев последовательного совпадения текущей точки с известными (опорными) точками поверхности:

$$\begin{cases} \lambda_1\varphi(r_{11})+\lambda_2\varphi(r_{12})+\cdots+\lambda_N\varphi(r_{1N})=z_1,\\ \lambda_1\varphi(r_{21})+\lambda_2\varphi(r_{22})+\cdots+\lambda_N\varphi(r_{2N})=z_2,\\ \vdots\\ \lambda_1\varphi(r_{N1})+\lambda_2\varphi(r_{N2})+\cdots+\lambda_N\varphi(r_{NN})=z_N, \end{cases} \tag{2}$$

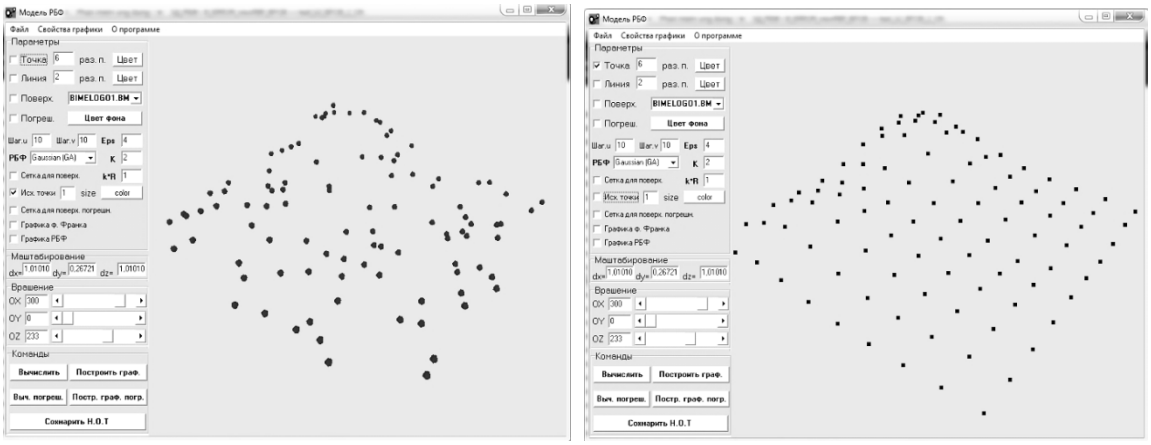
где r_{ij} – расстояние между i-й и j-й опорными точками.

Решение системы (2) возможно, например, методом Гаусса.

Первый этап моделирования выполняется с помощью программы «Модель РБФ», которая написана на языке C++ и позволяет определить новые опорные точки, равномерно расставленные в пространстве. В основу программы положен следующий алгоритм:

- получение координат исходных опорных точек;
- разбиение множества опорных точек на подмножества по заданному выбору;
- создание массива, хранящего координаты подмножества исходных опорных точек;
- нахождение коэффициентов опорной точки: создание и решение системы уравнений (2);
- вычисление координат новых опорных точек с помощью РБФ;
- сохранение координат новых опорных точек в массив и возврат к шагу 3 для обработки нового подмножества;
- запись данных новых опорных точек в Excel-файл;

Данные исходных опорных точек можно вводить с клавиатуры или загрузить из Excel-файла. Кроме того, пользователь может построить графики с помощью РБФ различного вида и определить их погрешность. Главное окно программы с примером приведено на рис. 1.



а) *б*)

Рис. 1. Окно программы «Модель РБФ» с исходным набором опорных точек (*а*) и с новым набором опорных точек (*б*)

Оценена погрешность интерполяции поверхности с применением РБФ и описанием интерполянта в явной форме. В качестве тестовой поверхности выбрана поверхность, описываемая функцией Франке [2]. Для оценки использованы следующие функции: гауссиан, мультиквадрик, инверсный мультиквадрик, квадратичная и кубическая функции Матерна, сплайны Вендланда [3, 4]. Они локализованы в пространстве и хорошо подходят для управления формой поверхности. За критерий оценки РБФ принято среднеквад-ратическое отклонение (СКО) реконструированной поверхности от тестовой. Наименьшее значение СКО получено для квадратичной функции Матерна. При шаге обхода 0,2, которому соответствует размерность сетки новых опорных точек в 46 шагов, величина СКО составляет 0.0019, что говорит о возможности применения описанного подхода.

Второй этап геометрического моделирования – переход к сеточной модели с применением интерполяции B-сплайнами. B-сплайны выбраны потому, что их отсеки легко стыкуются в гладкую поверхность. Для перехода к сплайновой модели необходимо обойти поверхность (1) с заданным шагом по обоим координатам-аргументам. Полученные значения x, y, z дают новые опорные точки поверхности. Они используются впоследствии в качестве вершин характеристических многогранников сплайновых отсеков. Их проекции на плоскость *xy* расставлены равномерно.

Традиционное параметрическое описание B-сплайнового отсека поверхности с номерами *i* и *j* по осям координат-параметров *u* и *v,* построенного на 16 опорных точках

$P_{i-1,j-1}$, $P_{i-1,j}$, …, $P_{i-1,j+2}$, $P_{i,j-1}$, …, $P_{i+2,j+2}$, которые заданы векторами координат $p_{p,q}=|x_{p,q}y_{p,q}z_{p,q}|$, где $p = i–1, i, i+1, i+2$ и $q = j–1, j, j+1, j+2$ в матричной форме имеет вид [3]:

$$R = U \cdot N \cdot P \cdot N^T \cdot V^T,$$

где $R = |xyz|$ – вектор координат текущей точки отсека поверхности;

$U = |u^3u^2u1|$ – вектор степеней параметра u, $u = [0, 1]$;

$V = |v^3v^2v1|$ – вектор степеней параметра v, $v = [0, 1]$;

P – матрица, элементами которой являются векторы координат опорных точек $p_{p,q}$;

N – базисная матрица B-сплайна:

$$N = \frac{1}{6}\begin{vmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{vmatrix}.$$

Если учесть равномерную расстановку опорных точек на плоскости аргументов, можно значительно упростить описание отсека поверхности. Оно примет следующий вид:

$$x = (1-u)x_{i,j} + ux_{i+1,j}, \quad (3)$$

$$y = (1-v)y_{i,j} + vy_{i,j+1}, \quad (4)$$

$$z = U \cdot N \cdot Z \cdot N^T \cdot V^T, \quad (5)$$

где Z – матрица высот опорных точек:

$$Z = \begin{vmatrix} z_{i-1,j-1} & z_{i-1,j} & z_{i-1,j+1} & z_{i-1,j+2} \\ z_{i,j-1} & z_{i,j} & z_{i,j+1} & z_{i,j+2} \\ z_{i+1,j-1} & z_{i+1,j} & z_{i+1,j+1} & z_{i+1,j+2} \\ z_{i+2,j-1} & z_{i+2,j} & z_{i+2,j+1} & z_{i+2,j+2} \end{vmatrix}.$$

Отсек рассматривается в своей локальной системе координат. Параметры u и v в данном случае определяют шаги вдоль каждого направления локальной системы координат. Эти направления совпадают с направлениями координатных осей x и y принятой декартовой системы координат поверхности. При изменении параметров в интервале [0,1] формируется один отсек поверхности. Для перехода к соседнему отсеку нужно заменить численные значения векторов $p_{m,n}$.

Отсек B-сплайна в общем случае не проходит через опорные точки, а только приближается к ним. Это свойство B-сплайна. Чтобы обеспечить точное прохождение поверхности через опорные точки, нужно еще раз изменить их расстановку. Нужно ввести новые опорные точки $P^*_{i,j}$,$P^*_{i,j+1}$,$P^*_{i+1,j}$,$P^*_{i+1,j+1}$,такие, что отсек поверхности пройдет через исходные точки $P_{i,j}$,$P_{i,j+1}$,$P_{i+1,j}$,$P_{i+1,j+1}$. Для одного отсека условием прохождения через исходные опорные точки является система из четырех уравнений, полученная подстановкой в выражения (3)–(5) пар значений $u = 0$, $v = 0$; $u = 0$, $v = 1$; $u = 1$,$v = 0$ и $u = 1$, $v = 1$. В общем случае на множестве, включающем $M \times N$ опорных точек, можно построить $(M–3)\times(N–3)$ отсеков поверхности. Для нахождения их $(M–2)\times(N–2)$ новых опорных точек нужно записать систему из $(M–2)\times(N–2)$ уравнений из условия прохождения поверхности через исходные опорные точки. Для каждого очередного отсека используются названные подстановки значений: $u = 0$, $v = 0$; $u = 0$, $v = 1$; $u = 1$, $v = 0$ и $u = 1$, $v = 1$. Кроме того, составная поверхность, состоящая из B-сплайновых отсеков, не проходит через крайние опорные точки, что при проектировании объектов может быть нежелательным. Чтобы поверхность проходила через все опорные точки, надо добавить несколько дополнительных крайних точек. Эти точки размещаются на небольшом расстоянии от крайних точек исходного множества, в том числе и в диагональных направлениях [4]. Таким образом, результатом предварительных вычислений является набор координат опорных точек, найденных из условий точного прохождения через них отсеков B-сплайна.

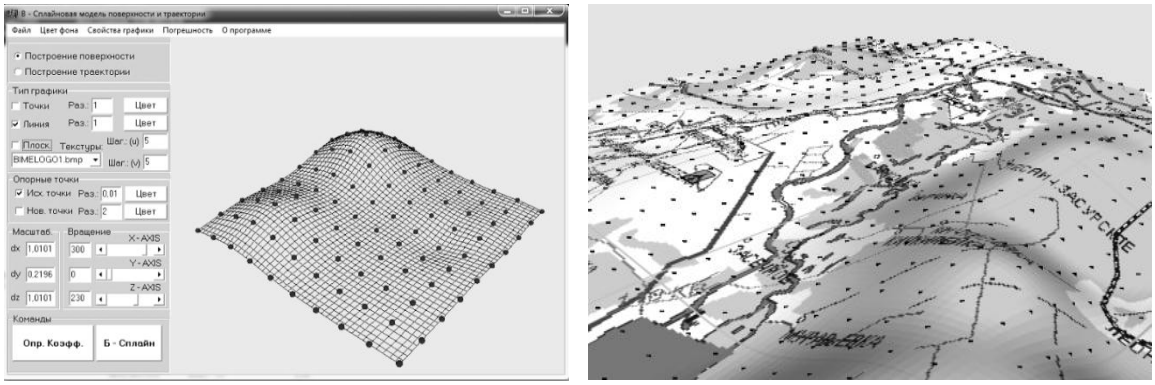
Нахождение текущей точки B-сплайновой поверхности предполагает вычисление в системе уравнений (3)–(5) двух линейных функций и одной бикубической функции параметров u и v. Выражение (5) содержит ресурсоемкие операции возведения в степень и перемножения. Для уменьшения вычислительной сложности применяются конечно-разностные формулы [5].

На втором этапе, для отображения пространственных форм используется программа «B-сплайновая модель траектории и поверхности», которая написана на языке C^{++} и

предназначена для компьютерного синтеза на экране монитора гладких криволинейных траекторий и поверхностей, характеризующихся большой протяженностью и точно проходящих через равномерно расставленные характерные точки. В основу программы положен следующий алгоритм:

- получение координат исходных опорных точек;
- добавление дополнительных точек;
- создание массива, хранящего координаты множества исходных опорных точек;
- поиск новых опорных точек: создание и решение системы уравнений, составленных из условия прохождения кривой, поверхности через исходные опорные точки;
- вычисление промежуточных значений функций-интерполянтов и их частных производных методом конечных разностей;
- сопоставление промежуточных точек и нормалей вершинам полилинии, полигональной сетки;
- построение кривой, поверхности на полученных вершинах средствами графической библиотеки.

Главное окно программы с примером построенной поверхности приведено на рис. 2. В левой части окна размещены средства ввода исходных данных и настройки параметров изображения, правая часть отведена под поверхности объектов с текстурой. В качестве текстуры поверхности используются фотографии или рисунки ландшафта. Ракурс обзора и масштаб поверхности объектов выбираются пользователем.



а) *б*)

Рис. 2. Результат интерполяции B-сплайнами с точным прохождением через новые опорные точки в виде сетки (*а*) и с текстурой (*б*) в программе «B-сплайновая модель траектории и поверхности»

Геометрическое моделирование пространственных форм для графических приложений необходимо выполнять в соответствии с требованиями визуализации. Эффективным решением является двухэтапная интерполяция. Оба этапа легко алгоритмизируются. Переход к упорядоченным опорным точкам позволяет применить быструю сеточную интерполяцию B-сплайнами**.** Для ускорения переходя к полигональной модели применяется конечно-разностное представление сплайн-функции. Разработанный программный комплекс позволяет исследовать возможности и выбирать значения параметров созданных моделей. Эксперимент показал хорошие изобразительные возможности описанного подхода.

Библиографический список

1. Kosnikov, Yu. N. Increasing the information capacity of the interface of the control system for multyparametric objects / Yu. N. Kosnikov // Application of Information and Communication Technologies – AICT2015 : 9th International conference: Conference Proceedings (14–16 October 2015). – Rostov-on-Don, Russia, 2015. – P. 446–449.

2. Franke, R. A Critical Comparison of Some Methods for Interpolation of Scattered Data. PhD thesis / R. Franke // Naval Postgraduate School Monterey. – California, 1979.

3. Kosnikov, Yu. N. Technique and technology of computer modeling of free forms surfaces with radial basis functions application / Yu. N. Kosnikov // XXI century: Resumes of the Past and Challenges

of the Present plus. Scientific Periodical Series: Engineering Sciences. Information Technologies. – 2014. – Vol. 1, № 03 (19). – P. 177–184.

4. Хоанг, Тхай Хо. Методика геометрического моделирования пространственных форм с применением интерполяции B-сплайнами / Хоанг Тхай Хо // Проблемы информатики в образовании, управлении, экономике и технике : сб. ст. XVI Междунар. науч.-техн. конф. – Пенза, 2016. – С. 70–78.

5. Косников, Ю. Н. Программная визуализация поверхностей заданных скалярным полям / Ю. Н. Косников, Хоанг Тхай Хо // Новые информационные технологии и системы : сб. науч. ст. XIII Междунар. науч.-техн. конф. – Пенза : Изд-во ПГУ, 2016. – С. 200–204.

А. Н. Токарев, А. В. Прошкин

Пензенский государственный университет, Пенза, Россия

МЕТОДИКА РАСЧЕТА ОБЪЕМОВ ПОДЗАДАЧ ПРИ РАВНОМЕРНОМ ДЕЛЕНИИ В РАСПРЕДЕЛЕННЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМАХ С НЕНАДЕЖНЫМИ УЗЛАМИ

Для распределенных вычислительных систем с ненадежными узлами необходима разработка такой методики расчета объемов подзадач, которая обеспечивала бы максимальную загрузку и минимальные потери от отказов узлов. При равномерном делении важнейшей задачей является нахождение объема одной подзадачи.

Наиболее часто для упрощения систем управления заданиями в кластерных и метакомпьютерных системах используется равномерное деление задачи на подзадачи. Однако, в системах с ненадежными узлами имеется вероятность потери результата при отказе узла, поэтому необходима выработка стратегии размещения подзадач, которая делила бы задачу на подзадачи такого объема, который, во-первых, был бы достаточно велик для того, чтобы не нагружать сеть и не терять производительность из-за латентности, и в то же время достаточно мал, чтобы не терять производительность из-за пересчета подзадач при отказе узлов.

При выработке такой стратегии будем использовать следующие предположения:

1) задача делится на подзадачи до начала расчета, и во время функционирования системы переразбиения задачи не происходит;

2) задача делится на равные по объему подзадачи;

3) подзадачи не привязаны к вычислительным узлам, и любой узел может осуществить расчет любой подзадачи;

4) узлы не являются идеально надежными, и для каждого узла имеется статистически полученная характеристика надежности.

Исходя из этого, можно сказать, что результатом, который должен быть получен в результате выработки данной стратегии, должно стать значение количества подзадач m, на которое равномерно поделится вся задача.

Поясним сначала на примере, почему необходимо уделить внимание делению задачу на подзадачи для такой системы.

а) Рассмотрим простейшую ситуацию, когда вычислительная система состоит из N клиентских машин одинаковой производительности, которые являются идеально надежными, т.е. сбой ни в одной из этих машин произойти не может (вероятность безотказной работы $P_{wi} = 1$, где $i = 1, 2, ..., N$).

В этом случае ясно, что оптимальным будет разбиение задачи на число подзадач, красное числу клиентских машин, т.е. N, $2N$, $3N$, и т.д.

Если не учитывать ситуацию, когда на сервере возникают очереди отправки подзадач, то при этом все клиенты получают исходные данные и отсылают результаты примерно в одно и то же время, и, таким образом, результат будет готов, как только все клиенты посчитают последний набор из N подзадач.

Какое же число подзадач выбрать для данного случая? Все будет определяться накладными расходами на пересылку данных по сети. Чем больше блок, тем меньше накладные расходы на заголовки пакетов и сопутствующие параметры пересылки, которые в конечном итоге определяют латентность. Таким образом, можно предположить, что идеальным вариантом будет разбиение на N подзадач одинаковой длины.

Однако, при большом физическом размере блока начинает играть роль вероятность образования очередей пакетов, ожидающих отправки/приема по сети (особенно для относительно медленных сетей с общей средой передачи, таких как Ethernet).

б) Приблизим задачу к более реальной обстановке. Пусть вычислительная система состоит из N вычислительных узлов одинаковой производительности, но существует вероятность выхода из строя каждого узла (т.е. вероятность безотказной работы узла $P_{wi} \le 1$, где $i = 1, 2, ..., N$).

Что же будет с вычислительным процессом в этом случае? Допустим, количество подзадач равно N, как в предыдущем случае. Все вычислительные узлы получили данные и начали расчет. Сервер запустил отсчет времени и перешел в состояние ожидания результатов. В соответствии с предположением, что узлы не являются надежными, вычисление на одном из узлов было прервано, допустим на узле 5. Все узлы через среднее время решения задачи $\bar{t}$ вернули результаты серверу, кроме узла 5.

При возврате последнего результата сервер в соответствии со своим алгоритмом функционирования проверяет список нерешенных подзадач, и видит, что этот список пуст. Однако, в списке решаемых подзадач есть подзадачи, результат решения которых еще не получен. Так как есть свободные узлы, сервер переносит нерешенную подзадачу 5 из списка решаемых подзадач в список нерешенных. При этом подзадача отдается на вычисление первому свободному узлу (а они все свободны), например, 1.

Пока узел 1 решает подзадачу за узел 5, все остальные узлы простаивают.

Таким образом, получается, что при делении задачи на N подзадач время решения с одним вышедшим из строя узлом удваивается.

Теперь рассмотрим вариант с количеством блоков, равным $2N$.

В этом случае узлы получают «половинные блоки», т.е. по отношению к предыдущему случаю подзадачи уменьшенного в 2 раза размера. На этапе решения первого набора из N подзадач клиент 5 выходит из строя. По прошествии среднего времени решения задачи $\bar{t}_2 = \dfrac{\bar{t}}{2}$ все узлы, кроме, естественно, узла 5, возвращают результаты серверу и получают $(N-1)$ следующих подзадач для расчета. После возврата результатов первому освободившемуся узлу отдается ожидающая решения вторая подзадача узла 5, а второму узлу – перенесенная из списка решаемых подзадач первая подзадача узла 5.

Наконец, по прошествии следующего промежутка времени $\bar{t}_2$ два вышеупомянутых узла присылают результат на сервер, и задача считается решенной.

Все это проиллюстрировано на рис. 1.

Получается, что для решения той же самой задачи, что и в первом случае, понадобится время уже не $2\bar{t}$, а $\dfrac{3}{2}\bar{t}$. Проведя подобные рассуждения, например, для случая с $4N$ подзадачами, можно убедиться, что время решения при одном отказавшем узле будет равно $\dfrac{5}{4}\bar{t}$. И так далее.

Таким образом, уменьшая размер подзадачи, мы уменьшаем штраф, который несет система за каждый отказавший узел. Но одновременно с этим увеличивается количество подзадач, а значит, увеличиваются накладные расходы на передачу блоков с данными по сети.

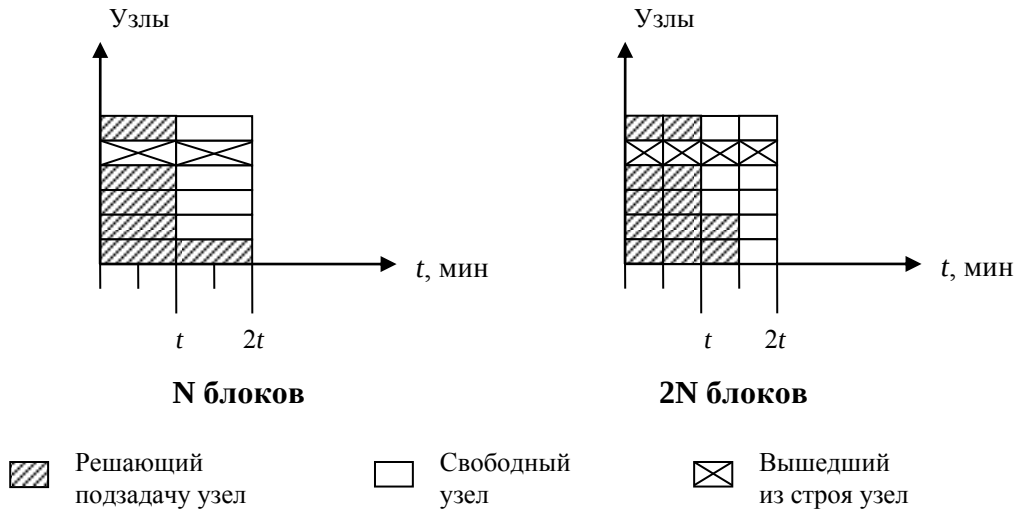


Рис. 1. Сравнение времени решения задачи для двух вариантов деления задачи при наличии отказавшего узла

Итак, основная задача заключается в том, чтобы с учетом характеристик надежности каждого узла определить, насколько можно увеличить размер блока данных, чтобы обеспечить минимальную совокупную латентность при сохранении высокой отказоустойчивости.

Из-за того, что каждый узел имеет собственную характеристику надежности, необходимо найти время безотказной работы, соответствующее узлу с минимальной надежностью и распределять подзадачи, ориентируясь на этот узел. Если этого не сделать, узлы, имеющие время безотказной работы меньшее, чем время решения подзадачи, будут с большой вероятностью отказывать, не решив свою подзадачу, и совокупная эффективность работы значительно снизится.

Для решения этой задачи нужно задать для всех узлов исходную вероятность решения подзадачи P_w, которая будет нас удовлетворять.

В соответствии с этой вероятностью по графику характеристики надежности каждого узла определяется время T_{wi}, по истечении которого вероятность отказа узла будет выше определенного нами порога.

В результате получим вектор времен $T_w = \{T_{w1}, T_{w2}, ..., T_{wN}\}$, по которому определяется другой вектор – вектор подзадач $S_i = \{S_1, S_2, ..., S_N\}$.

Вычисление объема каждой подзадачи производится по формуле нахождения объема блока (методика нахождения объема блока рассмотрена, например, в [1]):

$$S_i' = \frac{I_{Ci} \cdot I_{Ti} \cdot (t_{oi} - 2 \cdot t_{Li})}{k_C \cdot I_{Ti} + (k_S + k_R) \cdot I_{Ci}}$$

После этого определяется подзадача минимального объема:

$$S_{\min} = \min(S_i),\ i = 1, 2, ..., N.$$

Исходя из этого объема и определяется количество подзадач, на которое будет поделена вся задача перед началом расчета:

$$m = \frac{S}{S_{\min}}.$$

Необходимость рассчитывать вектор $S_i = \{S_1, S_2, ..., S_N\}$ возникает из-за того, что в рассматриваемом здесь случае мы считаем, что узлы имеют разную производительность. Если принять допущение, что все узлы одинаковы по своей производительности или их

производительность отличается незначительно, можно упростить процедуру нахождения m. При этом достаточно будет найти вектор $T_w = \{T_{w1}, T_{w2}, ..., T_{wN}\}$ и определить разбиение как

$$m = \sum_{i=1}^{N} \frac{T_{wi}}{\min(T_w)}$$

Нахождение разбиения в обоих случаях проиллюстрировано на рис. 2.

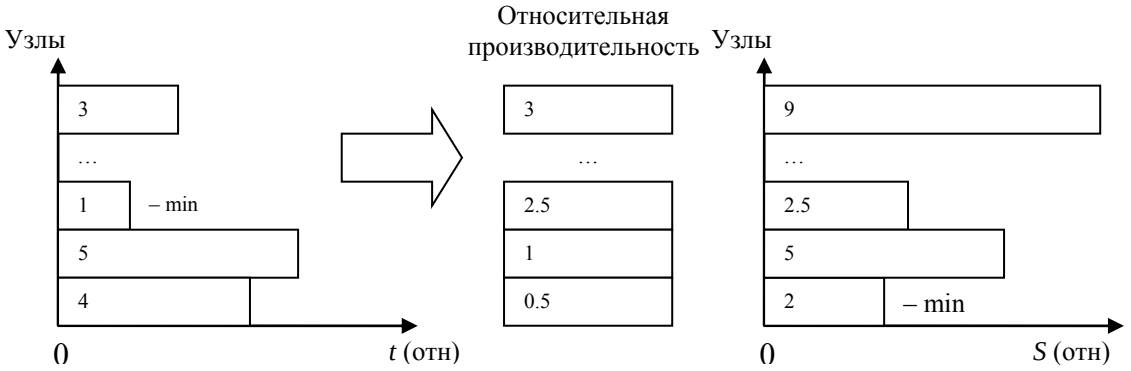


Рис. 2. Определение минимальной подзадачи при равномерном разбиении

Таким образом, предложенный подход позволяет осуществить эффективное равномерное разбиение задачи на подзадачи без использования большого количества исходных данных. Такое разбиение будет достаточно эффективным при решении несложных задач в распределенных системах с преимущественно однородной структурой.

Библиографический список

1. Токарев, А. Н. Стратегия размещения подзадач в распределенных вычислительных системах кластерно-метакомпьютерного типа : дис. … канд. техн. наук / Токарев А. Н. – Пенза : Изд-во ПГУ, 2005.

2. Токарев, А. Н. Оптимизация размещения подзадач в распределенных вычислительных системах с ненадежными узлами / А. Н. Токарев // Актуальные проблемы современной науки : тр. I Междунар. форума. – Самара, 2005. – С. 113–116.

Р. Н. Кузнецов, О. Ю. Кузнецова

Пензенский государственный университет, Пенза, Россия

НЕЙРОСЕТЕВАЯ ДИАГНОСТИКА ПОСЛЕОПЕРАЦИОННЫХ ОСЛОЖНЕНИЙ ПРИ ЖЕЛЧНОКАМЕННОЙ БОЛЕЗНИ

Представлены сведения о применении нейросетевого метода диагностики послеоперационных осложнений при желчнокаменной болезни (ЖКБ). Экспериментально было установлено, что производительность нейронной сети с архитектурой 5-9-2 составляет 87 %, что существенно не отличается от производительности сети на полном наборе.

На сегодняшний день желчнокаменная болезнь (ЖКБ) является наиболее распространенным заболеваниям в мире и занимает третье место после сахарного диабета и сердечно-сосудистых заболеваний [1–6]. Проблема своевременной диагностики и мониторин-

га послеоперационных осложнений у больных ЖКБ затруднена сложностью оценки степени тяжести состояния больного. Задача диагностики синдрома эндогенной интоксикации является на сегодняшний день актуальной:

1) 10–20 % взрослого населения России и Европы страдают ЖКБ [4–6];

2) 3,7 % до 37,3 % осложнений после открытых операций на желчевыводящих путях;

3) гнойные осложнения являются причиной длительного пребывания пациента в стационаре.

Работа основана на результатах обследования больных, которые находились в Областной клинической больнице им. Н. Н. Бурденко. Были отобраны 109 пациентов с ЖКБ, из них 63 случая без послеоперационных осложнений, 46 случаев с наличием осложнений. Для прогнозирования использовались общий и биохимический анализ крови.

По результатам статистического анализа были отобраны наиболее информативные признаки: Лейкоциты, Нейтрофилы палочкоядерные, Лимфоциты, Общий билирубин, Длительность оперативных вмешательств. Это позволило сократить число анализов с 15 до 5.

В ходе экспериментов проанализированы и построены различной конфигурации нейронные сети. Исследования проводились с использованием пакета Neural Network Toolbox системы MATLAB.

Нейронная сеть, реализующая прогнозирование послеоперационных осложнений у больных ЖКБ, решает задачу классификации. Входами сети являются пять отобранных показателей. Единственным выходом сети является признак «наличие – отсутствие» послеоперационных осложнений. Нейронная сеть обучалась по алгоритму Бройдена–Флетчера–Гольдфарба–Шанно (BFGS) [7]. BFGS является квазиньютоновским методом безусловной оптимизации второго порядка. Идея методов второго порядка заключается в разложении функционала качества работы нейронной сети f по формуле Тейлора:

$$f(p+x_k) \approx f(x_k) + \nabla f^T(x_k)\, p + \frac{1}{2} p^T H_k p,$$

где H – матрица Гессе функционала f в точке x, ∇f – вектор градиента функционала.

Квазиньютоновские методы являются итерационными и требуют решения на каждой итерации решения системы линейных алгебраических уравнений с матрицей Гессе. В BFGS вместо сложного вычисления матрицы Гессе и решения системы на каждой итерации вычисляется приближенное значение обратной матрицы Гессе.

В табл. 1–2 представлены характеристики качества работы архитектур нейронных сетей. Прогноз считался верными, если выход сети был более 0,8.

Таблица 1

Результаты обучения нейронных сетей с пятью входным нейронами

Архитектура	Ошибка (%)	AUC
5-3-2	63,16	0,71
5-11-2	71,05	0,74
5-9-2	86,84	0,92
5-15-2	76,31	0,67
5-10-2	64,47	0,71
5-13-2	63,15	0,74
5-17-2	60,52	0,69

Таблица 2

Результаты обучения нейронных сетей с шестнадцатью входным нейронами

Архитектура	Ошибка (%)	AUC
16-5-2	77,92	0,82
16-13-2	80,52	0,83
16-16-2	72,73	0,79
16-5-2	70,13	0,79

Анализ архитектур нейронных сетей при прогнозировании послеоперационных осложнений у больных ЖКБ показал, что прогнозирование на сокращенном наборе и прогнозирование на полном наборе имеют примерно одну и ту же ошибку, с той лишь разницей, что скрытый слой сетей содержал меньшее количество нейронов.

Также для оценки качества классификации использовали ROC-анализ, рассчитывался показатель AUC [8]. Показатель AUC (площадь под ROC-кривой) дает количественную интерпретацию ROC-кривой. Считается, что чем выше показатель AUC, тем качественнее классификатор. Из этого следует, что наиболее удачная архитектура нейронной сети является 5-9-2. Данная архитектура обучалась с использованием алгоритма BFGS, функцией активации скрытых нейронов являлся гиперболический тангенс, а функцией активации выходных нейронов являлась softmax.

Учитывая процент ошибки при прогнозировании, можно предположить, что применение нейронных сетей технологий для прогнозирования послеоперационных осложнений у больных ЖКБ поможет снизить время нахождения больных в стационаре.

Библиографический список

1. Головской, Б. В. Новый диагностический синдром при хроническом бескаменном холецистите / Б. В. Головской // Третий Всесоюзный съезд гастроэнтерологов : материалы съезда. – М., 1984. – Т. 1. – С. 238–239.

2. Ивашкин, В. Т. Клиника и диагностика функциональных запоров / В. Т. Ивашкин, Е. А. Полуэктова // Лечащий врач. – 2001. – № 5. – С. 26–29.

3. Лемешко, З. А. Возможности современных методов визуализации желчевыводящих путей и печени / З. А. Лемешко, В. Г. Никитин // Русский Медицинский журнал. – 1996. – Т. 4, № 3. – С. 25–28.

4. Гусева, В. А. Влияние смесей пищевых продуктов на двигательную функцию желчного пузыря при некоторых заболеваниях органов пищеварения : автореф. дис. … д-ра мед. наук / Гусева В. А. –М., 1968. – 24 с.

5. Ильченко, А. А. Желчнокаменная болезнь / А. А. Ильченко. – М. : Анахарсис, 2004. – 200 с.

6. Поляк, Е. З. Рентгенологические показатели основных функций желчного пузыря в норме и при холецистите : автореф. дис. … д-ра мед. наук / Поляк Е. З. – Киев, 1965. – 28 с.

7. Гилл, Ф. Практическая оптимизация / Ф. Гилл, У. Мюррей, М. Райт. – М. : Мир, 1985.

8. Паклин, Н. Б. Бизнес-аналитика: от данных к знаниям / Н. Б. Паклин, В. И. Орешков. – СПб. : Питер, 2013.

И. А. Абрамов, А. Н. Банникова, Н. А. Мартышкина

Пензенский государственный университет, Пенза, Россия

РЕАЛИЗАЦИЯ АЛГОРИТМА ПАРАЛЛЕЛЬНОГО ВЫДЕЛЕНИЯ КОНТУРОВ ОБЪЕКТОВ НА ИЗОБРАЖЕНИИ

Рассмаривается один из подходов ускорения выделения контуров объектов на изображении за счет многопроцессного выполнения, приводятся экспериментальные данные ускорения параллельной программы.

Обработка графической информации с использованием средств вычислительной техники применяется во многих сферах деятельности. Одной из основных задач обработки и анализа изображений является сегментация, в основе которой лежит разделение изображения на непересекающиеся области, покрывающие все изображение и однородные по некоторым признакам. Результатом сегментации является формирование карты областей изображения.

В настоящее время разработано достаточно много методов сегментации изображений. Одним из широко используемых методов является метод выделения контуров. Контурам изображения будут соответствовать перепады яркости, вызванных разницей уровней яркости фона и объекта. Для обнаружения перепада яркости используется значение градиента $\nabla\mathbf{f}(x,y)$ в точке (x,y), вычисление которого состоит в определении частных производных $\frac{\partial f}{\partial x}$ и $\frac{\partial f}{\partial y}$ точки цифрового изображения [1]. В этом случае предполагается, что изменение яркости точки, расположенной на границе изображения, должно быть значимо больше допустимого изменения яркости в точке фона. Способ определения значимости изменения яркости состоит в задании порога. Тогда точка будет является контурной, если ее первая производная превышает заданный порог.

При обнаружении контуров важную роль играет модуль вектора градиента, равный $\nabla f = |\nabla\mathbf{f}| = \sqrt{G_x^2 + G_y^2}$. Этот вектор с осью x составляет угол $\alpha(x,y)$, равный $\alpha(x,y) = \operatorname{arctg}\left(\frac{G_y}{G_x}\right)$. Тогда направление контура в точке (x,y) будет перпендикулярно направлению вектора градиента в этой точке [2].

Одним из способов нахождения частных производных $\frac{\partial f}{\partial x}$ и $\frac{\partial f}{\partial y}$ точки цифрового изображения состоит в применении оператора Собела. При этом изображение сканируется окном размером 3×3 (рис.) и вычисляются значения частных производных по формулам:

$$G_x = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)$$

и

$$G_x = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7),$$

где Z_i – значения окрестности на рис. 1.

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

Рис. 1. Окрестность 3×3 внутри изображения (переменные z_i представляют собой значение яркости)

В процессе выделения контуров обычно решается задача формирования из множества контурных точек границ объектов. Для этого выполняется анализ характеристик пикселов в окрестности сканирующего окна каждой точки изображения, которая помечена как контурная. Все точки, удовлетворяющие заранее заданным критериям, связываются и образуют контур. Пиксель контура, имеющий координаты (x_0, y_0) и расположенный внутри заданной окрестности точки (x,y), считается сходным по модулю градиента с пикселем (x,y), если

$$|\nabla f(x,y) - \nabla f(x_0,y_0)| \le E,$$

где E – заданный неотрицательный порог.

Пиксель контура, имеющий координаты (x_0, y_0) и расположенный внутри заданной окрестности точки (x,y), считается сходным по направлению градиента с пикселем (x,y), если

$$|\alpha(x,y) - \alpha(x_0,y_0)| \le A,$$

где A – заданный неотрицательный угловой порог.

Пиксель в заданной окрестности объединяется с центральным пикселем (x,y), если выполнены критерии сходства по величине и по направлению. Этот процесс повторяется в

каждой точке изображения, с одновременным запоминаем найденных связанных пикселей при движении центра окна. Простой способ учета данных состоит в том, что каждому множеству связываемых пикселей контура присваивается свое значение яркости [1].

Время обработки изображений существенно зависят от размеров самого изображения и маски. Можно приблизительно оценить время выполнения алгоритма. Определение значения градиента в пределах маски размера 3×3 с учетом трех составляющих цвета модели RGB потребуется выполнить 39 арифметических операций без учета операций управления циклом. Если принять, что τ_c – это время выполнения одной арифметической операции [3], то приблизительное время выполнения связывания контура внутри заданной окрестности точки равно $39\tau_c$. Тогда, с учетом размера изображения $N \times M$, полное время работы алгоритма на одном процессоре составит

$$T_1 = 39\tau_c N \times M.$$

Алгоритм связывания контуров требует выполнения множества однотипных операций, что свидетельствует о наличие параллелизма по данным. Как результат, распараллеливание может свестись к распределению данных между процессами, каждый из которых сможет независимо выполнить обработку своей части изображения. Для этого исходное изображение предполагается разбить на горизонтальные полосы и выделить каждому процессу подмножество строк в количестве N/p, где p – число создаваемых рабочих процессов, а N – размер изображения по вертикали [4].

Для оценки времени выполнения параллельного алгоритма можно предположить, что исходные данные уже распределены по процессорам нужным образом, а по окончании работы результат также распределен по процессорам. Тогда время выполнения алгоритма на системах с p процессоров будет равно $39\tau_c N \times M/p$. Это время учитывает только выполнение вычислительных операций, но не учитывает издержки на получение процессором данных и отправку результата. Пусть на выполнение одной операции передачи данных требуется время τ_s [3], а пересылка трех компонентов цвета (красного, зеленого и синего) составит $3\tau_s$. Тогда полное время для получение исходных данных одному процессу и отправки результата составит $6\tau_s N \times M/p$. Исходя из всего выше сказанного, время, затраченное каждым процессом, будет равно

$$T_p = \frac{3 \times N \times M(13\tau_c + 2\tau_s)}{p}.$$

Используя показатели T_1 и T_p можно определить, во сколько раз параллельный алгоритм будет выполняться быстрее последовательного.

$$S_p = \frac{T_p}{T_1} = \frac{N \times M \times 39\tau_c \times p}{3 \times N \times M(13\tau_c + 2\tau_s)} = \frac{13p}{13 + 2\frac{\tau_s}{\tau_c}}$$

Если принять время выполнения процессором арифметической операции примерно равным времени одной операции передачи или получения данных, то ускорение от распараллеливания алгоритма будет равно $S_p = 13p/15$. Для некоторых значений p ожидаемое ускорение алгоритма представлены в табл. 1.

Таблица 1

Приблизительные оценки ускорения алгоритма

Количество вычислительных узлов в системе	Ускорение
2	1,73
4	3,46
6	5,2
8	6,93

Для экспериментальной оценки ускорения параллельного алгоритма были разработаны последовательная и параллельная программы. Последовательная программа была реализована на языке C++ и скомпилированы с помощью компилятора gcc 5.4.0. Параллельная программа также реализована на языке C++с помощью технологии MPI. Эксперименты

с программой проводились на компьютерах на базе процессора Intel Core i3, 3,5 ГГц, 8 Гб ОЗУ под управлением операционной системы Ubuntu 16.04, которые объединены сетью Gigabit Ethernet В эксперименте были исследованы последовательная программа, двухпроцессный, четырехпроцессный, шестипроцессный и восьмипроцессный варианты выполнения параллельной программы.

В процессе эксперимента было обработано с помощью маски 3×3 изображение размером 5000×3000. Результаты вычислительных экспериментов выделения контуров на изображении приведены в табл. 2. Время выполнения указаны в миллисекундах.

Таблица 2

Результаты вычислительных экспериментов

Последовательный алгоритм, время, мс	Параллельный алгоритм							
	2 процесса		4 процесса		6 процессов		8 процессов	
	Время, мс	Ускорение	Время, мс	Ускорение	Время, мс	Ускорение	Время, мс	Ускорение
51186	30720	1,66	15632	3,27	10196	5,02	7618	6,71

Как видно из результатов эксперимента, ускорение параллельной программы немного ниже ожидаемого (см. табл. 1). Это может быть объяснено бо́льшими издержками на взаимодействие процессов, чем предполагалось. Тем не менее, использование распараллеливания позволило сократить время выделения контуров на изображении.

Библиографический список

1. Гонсалес, Р. Цифровая обработка изображений / Р. Гонсалес, Р. Вудс. – М. : Техносфера, 2005. – 1072 с.

2. Красильников, Н. Н. Цифровая обработка 2D- и 3D-изображений : учеб. пособие / Н. Н. Красильников. – СПб. : БХВ-Петербург, 2011. – 608 с.

3. Якобовский, М. В. Введение в параллельные методы решения задач : учеб. пособие / М. В. Якобовский ; предисл.: В. А. Садовничий. – М. : Изд-во Моск. ун-та, 2013. – 328 с.

4. Гергель, В. П. Современные языки и технологии параллельного программирования : учеб. / В. П. Гергель ; предисл.: В. А. Садовничий. – М. : Изд-во Моск. ун-та, 2013. – 408 с.

М. М. Алкезуини

Пензенский государственный университет, Пенза, Россия

СОВЕРШЕНСТВОВАНИЕ АЛГОРИТМОВ ГРАДИЕНТНОГО СПУСКА ОБУЧЕНИЯ СЕТЕЙ РАДИАЛЬНЫХ БАЗИСНЫХ ФУНКЦИЙ[1]

Для обучения сетей радиальных базисных функций впервые разработаны алгоритмы на основе методов первого порядка: градиентный спуск с импульсом, алгоритм ускоренного градиента Нестерова и RMSProp в сочетании с ускоренным градиентом Нестерова. Создан комплекс программ в системе MATLAB, реализующий разработанные алгоритмы. Проведены экспериментальные исследования разработанных алгоритмов.

Рассмотрим работу сети радиальных базисных функций (РБФ-сети) на примере решения задач аппроксимации. РБФ-сеть – это двухслойная сеть нейронная сеть, первый

[1] Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 16-08-00906.

слой которой состоит из радиальной базисной функции (РБ-функции), а второй является линейным сумматором [1]. В случае аппроксимации функции двух переменных сеть имеет два входа – координаты точки и один выход – значение функции в точке. Выход РБФ-сети описывается выражением

$$u(\mathbf{x}) = \sum_{k=1}^{n_{RBF}} w_k \varphi_k(\mathbf{x}), \quad (1)$$

где n_{RBF} – количество РБ-функций (количество нейронов), w_k – вес k-го нейрона, где $\mathbf{x}$ – входной вектор $\varphi_k(\mathbf{x})$ – значение k-ой РБ-функции в точке $\mathbf{x}$.

В данной работе в качестве РБ-функции использована функция Гаусса (гауссиан), в двумерном случае имеющая вид

$$\varphi(\mathbf{x}) = \exp\left(-\frac{\|\mathbf{x}-\mathbf{c}\|^2}{2a^2}\right), \quad (2)$$

где $\mathbf{c} = [c_1, c_2]^{\mathrm{T}}$ – вектор координат центра РБ-функции, a – ширина (параметр формы), $\|\mathbf{x}-\mathbf{c}\| = \sqrt{(x_1-c_1)^2-(x_2-c_2)^2}$ – евклидова норма (расстояние между точкой $\mathbf{x}$ и центром $\mathbf{c}$ РБ-функции).

При решении задач аппроксимации функций входные векторы при обучении сети представляют собой координаты пробных точек (узлов интерполяции), в двумерном случае $\mathbf{x} = [x_1, x_2]^{\mathrm{T}}$, а целевые значения – значения функции в пробных точках. После обучения сеть при подаче координат произвольной точки области определения функции выдает значение функции в этой точке.

Наличие только двух слоев упрощает обучение РБФ-сети. Но в процессе обучения необходимо настраивать две группы параметров сети: веса, которые входят линейно в выходной сигнал сети (1), и параметры РБ-функций (в случае гауссиана (2) – центры и ширина), которые входят нелинейно в выходной сигнал.

Обучение нейронной сети представляет собой минимизацию некоторого функционала ошибки. Используем функционал ошибки

$$I = \frac{1}{2}\sum_{j=1}^{n} e_j^2 = \frac{1}{2}\sum_{j=1}^{n}\left(u(\mathbf{p}_j - T_j)\right)^2, \quad (3)$$

где n – количество пробных точек, e_j – ошибка решения в j-й пробной точке, $\mathbf{p}_j$ – координаты j-й пробной точки (в случае аппроксимации функции двух переменных $\mathbf{p}_j = [p_{j1}, p_{j2}]^{\mathrm{T}}$), $u(\mathbf{p}_j)$ – решение (1) в j-й пробной точке, T_j – целевое значение в j-й пробной точке, множитель $1/2$ введен для упрощения вычислений

Для обучения РБФ-сетей используются, в основном, градиентные методы [1]. Градиентные методы делятся на методы первого порядка, использующие первые производные минимизируемой функции, то есть градиент функции, и методы второго порядка, использующие вторые производные – матрицу Гессе. Все градиентные алгоритмы позволяют найти только локальный минимум функции. Методы первого порядка просты в реализации, но работают медленно. Методы второго порядка выполняются за гораздо меньшее число итераций, но сложны и ресурсоемки. Для обучения РБФ-сетей используется простейший метод первого порядка – метод градиентного спуска. Известные методы второго порядка, например, метод Левенберга-Марквардта [1] не применяются для обучения РБФ-сетей, а предложенный в [2] быстрый алгоритм метода доверительных областей довольно сложен.

В настоящее время для глубокого обучения сетей, содержащих большое количество слоев и обучающихся на больших объемах данных, усилился интерес к несложным уско-

ренным градиентным методам первого порядка [3]. Такие методы не применялись для обучения РБФ-сетей. Целью настоящей работы является разработка и экспериментальное исследование ускоренных градиентных методов первого порядка для обучения РБФ-сетей при решении задач аппроксимации.

Рассмотрим алгоритм градиентного спуска. Если $\boldsymbol{\theta}$ – вектор одного или всех параметров сети (в нашем случае это векторы весов, центров и ширины РБ-функций), то настройка вектора $\boldsymbol{\theta}$ на k-й итерации описывается выражением

$$\boldsymbol{\theta}^{(k+1)} = \boldsymbol{\theta}^{(k)} + \Delta\boldsymbol{\theta}^{(k+1)}, \tag{4}$$

где $\Delta\boldsymbol{\theta}^{(k+1)} = -\eta \mathbf{g}_{\boldsymbol{\theta}}\left(\boldsymbol{\theta}^{(k)}\right)$ – поправка вектора $\boldsymbol{\theta}$, η – подбираемый числовой коэффициент (скорость обучения), $\mathbf{g}_{\boldsymbol{\theta}}\left(\boldsymbol{\theta}^{(k)}\right)$ – вектор градиента функционала I (3) по значению параметра $\boldsymbol{\theta}^{(k)}$ на итерации k. Процесс вычислений по (4) продолжается до малого значения функционала ошибки (3) или малого значения среднеквадратической погрешности

$$I_{MSE} = \frac{1}{n}\sum_{j=1}^{n}\left(u\left(\mathbf{p}_j\right) - T_j\right)^2. \tag{5}$$

В алгоритме градиентного спуска с импульсом (название происходит от аналогии с импульсом в механике) поправка к вектору параметров имеет вид

$$\Delta\boldsymbol{\theta}^{(k+1)} = \alpha\Delta\boldsymbol{\theta}^{(k)} - \eta\mathbf{g}_{\boldsymbol{\theta}}\left(\boldsymbol{\theta}^{(k)}\right), \tag{6}$$

где η – скорость обучения, α – коэффициент момента, принимающий значения в интервале [0, 1].

Наличие в (6) слагаемого, не зависящего от градиента, позволяет методу преодолевать плоские участки функционала ошибки !по инерции!.

В методе ускоренного градиента Нестерова (NAG – Nesterov Accelerated Gradient) [3] поправка вектора параметров описывается следующим образом

$$\Delta\boldsymbol{\theta}^{(k+1)} = \alpha\Delta\boldsymbol{\theta}^{(k)} - \eta\mathbf{g}_{\boldsymbol{\theta}}\left(\boldsymbol{\theta}^{(k)} + \alpha\Delta\boldsymbol{\theta}^{(k)}\right).$$

Метод NAG отличается от градиентного спуска с импульсом более точным вычислением вектора поправки.

При обучении глубоких сетей получили распространение алгоритмы с адаптивной скоростью обучения [3]. В этих алгоритмах используется различная скорость обучения для различных компонентов вектора параметров, так как разные компоненты могут оказывать различное влияние на обучение сети. Популярным алгоритмом этого типа является RMSProp (Root Mean Square Propagation) и его сочетание с ускоренным градиентом Нестерова [3], k-я имеет вид

$$\mathbf{g} = \mathbf{g}_{\boldsymbol{\theta}}\left(\boldsymbol{\theta}^{(k)} + \alpha\Delta\boldsymbol{\theta}^{(k)}\right), \quad \mathbf{r}^{(k+1)} = \rho\mathbf{r}^{(k)} + \left(1-\rho\right)\mathbf{g}\bullet\mathbf{g},$$

$$\Delta\boldsymbol{\theta}^{(k+1)} = \alpha\Delta\boldsymbol{\theta}^{(k)} - \frac{\varepsilon}{\sqrt{\mathbf{r}^{(k+1)}}}\bullet\mathbf{g}, \quad \boldsymbol{\theta}^{(k+1)} = \boldsymbol{\theta}^{(k)} + \Delta\boldsymbol{\theta}^{(k+1)},$$

где $\mathbf{r}^{(0)} = \mathbf{0}$, $\bullet$ – операция поэлементного умножения (произведение Адамара), $\sqrt{\mathbf{r}^{(k+1)}}$ вычисляется поэлементно, $\rho, \alpha, \varepsilon$ – подбираемые коэффициенты.

При обучении РБФ-сети можно реализовать одновременную и последовательную стратегию настройки векторов весов, центров и ширины на каждом цикле обучения. При одновременной настройке в каждой итерации при значениях векторов параметров, полученных на предыдущей итерации, находятся векторы градиентов функционала ошибки по

каждому параметру, затем вычисляются поправки параметров и вычисляются обновленные векторы параметров. При последовательной настройке сначала вычисляется градиент по весам (настройка весов оказывает наибольшее влияние на обучение сети) и вычисляется обновленный вектор весов. Затем с использованием обновленного вектора весов вычисляется градиент по центрам и вычисляется обновленный вектор центров. Наконец, с использованием обновленных векторов весов и центров вычисляется градиент по ширине и обновляется вектор ширины. Так как при последовательной настройке каждый параметр настраивается при обновленной части других параметров, можно предположить, что последовательная настройка может обеспечить сокращение числа итераций обучения.

Компоненты градиента функционала по весам, центрам и ширине несложно вычислить аналитически (рассмотрен случай аппроксимации функции двух переменных, номера итераций опущены)

$$\frac{\partial I}{\partial w_i} = \sum_{j=1}^{n}\left(u\left(\mathbf{p}_j\right)-T_j\right)\cdot\varphi_i\left(\mathbf{p}_j\right), \quad \frac{\partial I}{\partial c_{i1}} = w_i\sum_{j=1}^{n}\left(u\left(\mathbf{p}_j\right)-T_j\right)\cdot\varphi_i\left(\mathbf{p}_j\right)\cdot\frac{p_{j1}-c_{i1}}{a_i^2},$$

$$\frac{\partial I}{\partial c_{i2}} = w_i\sum_{j=1}^{n}\left(u\left(\mathbf{p}_j\right)-T_j\right)\cdot\varphi_i\left(\mathbf{p}_j\right)\cdot\frac{p_{j2}-c_{i2}}{a_i^2}, \quad \frac{\partial I}{\partial a_i} = w_i\sum_{j=1}^{n}\left(u\left(\mathbf{p}_j\right)-T_j\right)\cdot\varphi_i\left(\mathbf{p}_j\right)\cdot\frac{\left\|\mathbf{p}_j-\mathbf{c}_i\right\|}{a_i^3},$$

где c_{i1} и c_{i2} – координаты центра i-й РБ-функции, p_{j1} и p_{j2} – координаты пробной точки $\mathbf{p}_j$, $\left\|\mathbf{p}_j-\mathbf{c}_i\right\|$ – евклидова норма.

Экспериментальное исследование рассмотренных методов проводилось на примере аппроксимации функции $z=x^2+y^2$ в области $\left(x=-3\ldots+3,\ y=-3\ldots+3\right)$. Количество узлов интерполяции равно 100. Узлы интерполяции располагались случайным образом в области (рис. 1). Количество РБ-функций равно 16. В начальном состоянии центры РБ-функций располагались на сетке (рис. 1). Веса инициировались случайными числами, равномерно распределенными от 0 до 0,001. Начальная ширина всех РБ-функций была постоянной, равной 3,0. Итерационный процесс обучения продолжался до достижения значения средней квадратической ошибки (5), равной 0,01. Для проведения экспериментов разработан комплекс программ в системе MATLAB. Эксперименты проводились на компьютере со следующими характеристиками: процессор Intel Core i5-2500K, 3,30 GHz, ОЗУ 8,0 GB.

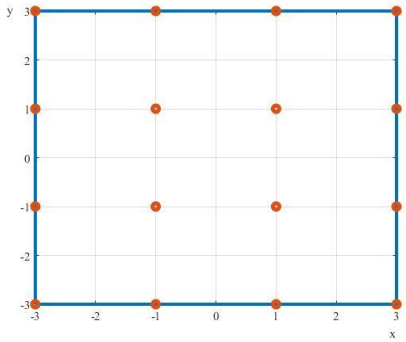


Рис. 1. Начальное расположение центров РБ-функций

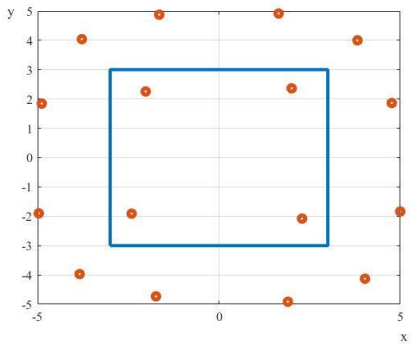


Рис. 2. Пример конечного положения центров РБ-функций

Результаты экспериментов представлены в таблице. Так как число итераций и время решения зависят от случайных начальных значений весов, то в таблице показаны диапазоны числа итераций и времени решения, полученные в 10 экспериментах. Индексы у коэффициентов означают: 1 – коэффициент для весов, 2 – для центров, 3 – для ширины. Значения коэффициентов подбирались экспериментально.

Результаты экспериментов показали явное преимущество ускоренных методов первого порядка по сравнению с классическим методом градиентного спуска. Последовательная настройка параметров также дает существенный эффект. Градиентный спуск с импуль-

сом очень чувствителен к случайным начальным значениям весов и параметрам обучения. Наиболее быстрым и устойчивым является метод NAG. Метод наименее чувствителен к начальным значениям весов и параметрам обучения. Экспериментально обнаружена необходимость использования различных значений коэффициентов α для различных параметров сети. Метод RMSProp+NAG устойчив к изменению начальных значений весов и параметров обучения, Но метод проигрывает NAG по числу итераций.

Таблица

Результаты экспериментов

Метод	Стратегия обучения	Параметры	Число итераций	Время решения, С
Градиентный спуск	Одновременная настройка	$\eta_1 = 0,00150$, $\eta_2 = 0,00100$, $\eta_3 = 0,00050$	45000–70000	2500–5000
Градиентный спуск	Последовательная настройка	$\eta_1 = 0,05000$, $\eta_2 = 0,00100$, $\eta_3 = 0,00050$	35000–50000	2000–3500
Градиентный спуск с импульсом	Одновременная настройка	$\eta_1 = 0,00700, \quad \alpha_1 = 0,9$, $\eta_2 = 0,00002, \quad \alpha_2 = 0,9$, $\eta_3 = 0,00020, \quad \alpha_3 = 0,9$	1500–2100	50–70
Градиентный спуск с импульсом	Последовательная настройка	$\eta_1 = 0,00700, \quad \alpha_1 = 0,9$, $\eta_2 = 0,00002, \quad \alpha_2 = 0,9$, $\eta_3 = 0,00020, \quad \alpha_3 = 0,9$	200–9000	9–360
NAG	Последовательная настройка	$\eta_1 = 0,00500, \quad \alpha_1 = 0,9$, $\eta_2 = 0,00200, \quad \alpha_2 = 0,5$, $\eta_3 = 0,00100, \quad \alpha_3 = 0,3$	270–470	14–24
RMSProp+NAG	Последовательная настройка	$\eta_1 = 0,00100$, $\eta_2 = 0,00200$, $\eta_3 = 0,00100$, $\alpha_1 = 0,90000, \quad \rho_1 = 0,90000$ $\alpha_2 = 0,50000, \quad \rho_2 = 0,90000$ $\alpha_3 = 0,10000, \quad \rho_3 = 0,90000$	5700–13400	245–600

Эксперименты подтвердили важность настройки не только весов, но и параметров РБ-функций. Конечное положение центров РБ-функций, полученное в результате обучения сети (рис. 2) радикально отличается от начального положения (рис. 1), причем после обучения сети центры вышли за пределы области решения.

Библиографический список

1. Осовский, С. Нейронные сети для обработки информации / С. Осовский. – М. : Горячая линия-Телеком, 2016. – 448 с.

2. Горбаченко, В. И. Решение краевых задач математической физики с помощью сетей радиальных базисных функций / В. И. Горбаченко, М. В. Жуков // Журнал вычислительной математики и математической физики. – 2017. – Т. 57, № 1. – С. 133–143.

3. Гудфеллоу, Я. Глубокое обучение / Я. Гудфеллоу, И. Бенджио, А. Курвилль. – М. : ДМК Пресс, 2017. – 652 с.

А. Ф. Валько, А. А. Валько

Пензенский государственный университет, Пенза, Россия

СТРУКТУРА ОПТИЧЕСКИХ ИЛЛЮЗИЙ В НЕПОДВИЖНЫХ ИЗОБРАЖЕНИЯХ ГРАФИЧЕСКИХ ОБЪЕКТОВ

Предложен способ применения свойств графических объектов для корректировки оптического восприятия данных объектов. Рассмотрен способ послойной реализации и хранения иллюзии как метод изменения восприятия.

Наличие в графических объектах специфических свойств, вызывающих искажение восприятия, позволяет использовать их для корректировки оптического восприятия изображений [1].

Свойства восприятия можно определить через принципы гештальтизма: фигура, фон, константы, которые соотносятся между собой и вместе образуют новое свойство [3]. Целостность восприятия обеспечивается такими свойствами как близость (объекты расположенные рядом воспринимаются вместе), схожесть (объекты, имеющие схожую форму, цвет воспринимаются вместе), целостность (восприятие объектов стремится к целостности и упрощению), замкнутость (определяет тенденцию к завершению фигуры или контура), смежность (близость объектов), общая зона (определяет восприятие наравне с научением и прошлым опытом).

В настоящее время геометрические оптические иллюзии могут быть использованы при разработке графических интерфейсов программных продуктов, в архитектуре и рекламе. Свойства оптических иллюзий, образованных графическими объектами и изображениями могут быть сгруппированы по следующим свойствам: зрительные искажения, иллюзии восприятия размера, иллюзии цвета и контраста, эффекты последействия, иллюзии движения, эффект перцептивной готовности, распознавание образов, следящие картинки, соотношение фигуры и фона, кажущиеся фигуры, иллюзии восприятия глубины, невозможные фигуры, перевернутые картинки, парейдолические иллюзии [2]. В свою очередь внутри групп оптические иллюзии образуются комбинацией взаимного расположения графических примитивов, градиентов цвета, размера и формы.

Для поиска элементов графических объектов на изображениях, являющихся оптическими иллюзиями, или элементов, из которых оптическую иллюзию можно скомпоновать, необходимо провести сопоставление графических примитивов и типов иллюзий их образующих.

Для организации процедуры распознания и анализа структуры объектов иллюзий можно использовать два подхода: первый подход связан с использованием нейронных сетей и процедур распознания элементов иллюзий, второй с организацией баз данных графических объектов иллюзий и применением специализированных алгоритмов распознавания образов.

Удобной формой хранения и обработки графических объектов является послойное представление. Такое представление обеспечивает удобство послойного анализа и редактирования каждого из типов иллюзий, например средствами Photoshop. Часть объектов каждого слоя может быть реализована, например, в виде контуров инструментами Photoshop. Но наиболее удобным является послойное векторное представление объектов оптических иллюзий.

Послойная реализация иллюзий, содержащихся на изображении, подразумевает иерархию не только типов иллюзий, но и послойную топологию структуры каждой отдельной иллюзии. Послойная топология содержит отдельные графические примитивы, привязанные к топологическим координатам объектов. Разложение каждого из типов иллюзий на примитивы определяется возможностями его влияния на восприятие. Кроме того большое значение имеет определение векторов направлений перемещения примитивов.

Каждый тип оптических иллюзий идентифицируется набором примитивов и стратегией изменения. Например, изменением толщины линий, интенсивностью цвета или вза-

имным расположением объектов, составляющих оптическую иллюзию. Данная структура представлена на рис. 1.

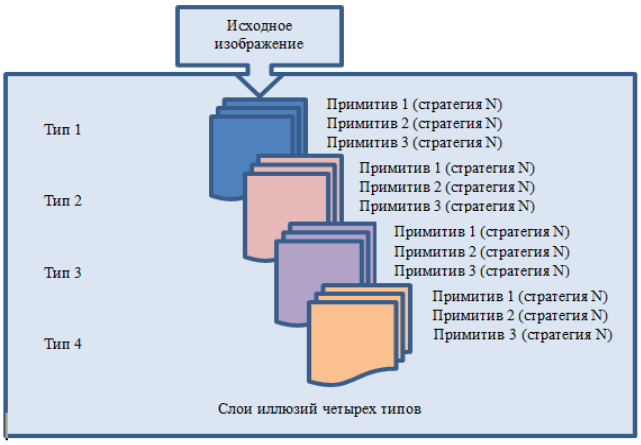


Рис. 1 Консолидация типов оптических иллюзий из N стратегий корректировок по слоям

Интенсивность восприятия иллюзии определяется параметрами слоя и взаимным расположением объектов, формирующих иллюзию. Часто один слой содержит, части объектов, участвующих в иллюзиях других слоев, что приводит к ограничениям при координатной манипуляции объектов. Наиболее простым подходом является исключение слоя из области видимости либо организации топологических связей объектов.

У каждой группы иллюзий изменение степени восприятия реализуется различными методами. Кроме того, нельзя получить усиление восприятия простым масштабированием объектов. Зачастую некоторые типы иллюзий, присутствующие на одной картине, могут ослаблять или усиливать степень восприятия общей иллюзорности, это накладывает ограничения на выбор слоя и компоновку графических примитивов и заливок, формирующих иллюзию.

Библиографический список

1. Двойственные миры Эшера. – URL: http://subscribe.ru/archive/
2. Оптические иллюзии и их применение. – URL: http://www.psy.msu.ru/illusion/
3. Целостность восприятия. – URL: http://psyznaiyka.net/

О. Э. Яремко

Пензенский государственный университет, Пенза, Россия

РЕШЕНИЕ ПЛОСКИХ ЗАДАЧ ТЕОРИИ УПРУГОСТИ СРЕДСТВАМИ MATLAB

Рассматривается упругое равновесие полубесконечного однородного тела в состоянии плоской деформации под действием усилий, приложенных к ограничивающей плоскости. Пакет Curve Fitting Toolbox из MATLAB применялся для реализации метода фундаментальных базисных функций при вычислении компонент тензора напряжений.

Теория упругости – это математическое исследование деформации твердых объектов и внутренних напряжений в связи с заданными условиями нагрузки. Из частных задач обобщенной плоской деформации одной из простейших является задача о распределении

напряжений в упругом однородном полупространстве под действием усилий, приложенных к ограничивающей плоскости и зависящих только от одной координаты. Рассмотрим упругое равновесие полубесконечного однородного тела, ограниченного плоскостью (упругое полупространство), в состоянии обобщенной плоской деформации под действием усилий, приложенных к ограничивающей плоскости.

Предположим, что на поверхности тела имеется давление $p = f(y)$. Тогда можно записать граничные условия

$$\sigma_x = -f(y), \tau_{xy} = 0 \text{ при } x = 0. \quad (1)$$

Обозначим интеграл Пуассона через

$$P_f = \frac{1}{2\pi}\int_{-\infty}^{\infty} \frac{x}{x^2 + (y-\eta)} f(\eta) d\eta. \quad (2)$$

Интеграл Пуассона дает решение задачи Дирихле для правой полуплоскости [1]. В монографии Снедонна [2, с. 455] представлены аналитические формулы для нормальных и касательных напряжений $\sigma_x, \sigma_y, \tau_{xy}$. Данными формулами можно пользоваться только в случае аналитического задания давления $p = f(y)$. В случае численного задания давления $p = f(y)$ актуальна проблема получения алгоритма решения плоских задач теории упругости, не требующего вычислений интегралов Пуассона. В [3] рассмотрено решение краевых задач с использованием сетей радиальных базисных функций. Поставим целью работы: для модельных задач плоской теории упругости разработать алгоритм решения, использующий обратные квадрики [4] в качестве радиальных базисных функций. Таким образом, радиально базисные функции конструируются из фундаментальных решений бигармонического уравнения.

Из формул, приведенных в монографии Снедонна [2], следуют выражения для компонент тензора напряжений

$$\sigma_x = -P_f + x\left(P_f\right)'_x, \sigma_y = -P_f - x\left(P_f\right)'_x, \tau_{xy} = x\left(P_f\right)'_y, \quad (3)$$

где P_f – интеграл Пуассона.

Обратной квадрикой [4] называют функцию вида

$$\frac{1}{b^2 + (y-c)^2},$$

где b, c – ширина и центр квадрики, соответственно.

Будем аппроксимировать давление $p = f(y)$ взвешенными суммами обратных квадрик,

$$f(y) \approx \sum_{k=1}^{N} w_k \frac{1}{b_k^2 + (y-c_k)^2}. \quad (4)$$

Интеграл Пуассона имеет вид,

$$P_f \approx \sum_{k=1}^{N} \frac{w_k}{b_k} \frac{(x+b_k)}{(x+b_k)^2 + (y-c_k)^2}.$$

На основе формул (4), получим выражения для компонент тензора напряжений

$$\begin{aligned}
\sigma_{xapr} &\approx -\sum_{k=1}^{N} \frac{w_k}{b_k} \frac{(x+b_k)}{(x+b_k)^2 + (y-c_k)^2} - x.\sum_{k=1}^{N} \frac{w_k}{b_k} \frac{(x+b_k)^2 - (y-c_k)^2}{\left((x+b_k)^2 + (y-c_k)^2\right)^2}, \\
\sigma_{yapr} &\approx -\sum_{k=1}^{N} \frac{w_k}{b_k} \frac{(x+b_k)}{(x+b_k)^2 + (y-c_k)^2} + x.\sum_{k=1}^{N} \frac{w_k}{b_k} \frac{(x+b_k)^2 - (y-c_k)^2}{\left((x+b_k)^2 + (y-c_k)^2\right)^2}, \\
\tau_{xyapr} &\approx -2\sum_{k=1}^{N} \frac{w_k}{b_k} \frac{(x+b_k)(y-c_k)}{\left((x+b_k)^2 + (y-c_k)^2\right)^2}.
\end{aligned} \quad (5)$$

Тестовая функция выбиралась в виде

$$p(y)=\operatorname{arctg}(y+1)-\operatorname{arctg}(y-1).$$

При таком значении давления компоненты тензора напряжений имеют вид

$$\sigma_{xtheor}=-\operatorname{arctg}\frac{y+1}{x+1}+\operatorname{arctg}\frac{y-1}{x+1}+x\left(-\frac{(y+1)}{(x+1)^2+(y+1)^2}+\frac{(y-1)}{(x+1)^2+(y-1)^2}\right),$$

$$\sigma_{ytheor}=-\operatorname{arctg}\frac{y+1}{x+1}+\operatorname{arctg}\frac{y-1}{x+1}+x\left(\frac{(y+1)}{(x+1)^2+(y+1)^2}-\frac{(y-1)}{(x+1)^2+(y-1)^2}\right),$$

$$\tau_{xtheor}=x\left(-\frac{(x+1)}{(x+1)^2+(y+1)^2}+\frac{(x+1)}{(x+1)^2+(y-1)^2}\right).$$

Вопросы использования систем компьютерной математики MAPLE, MATLAB в теории упругости изложены в [5–6].

Для аппроксимации давления $p=f(y)$ был использован пакет Curve Fitting Toolbox из MATLAB. Основные возможности пакета:

1. Графическое отображение исследуемых данных.
2. Сглаживание и фильтрование данных различными способами.
3. Приближение данных при помощи параметрических моделей, в которых искомые параметры могут входить как линейно, так и нелинейно
4. Отображение построенных приближений графически, форматирование графиков и сохранение результатов в отдельных графических окнах.
5. Выполнение операций с полученными приближениями (вычисление в заданных точках, дифференцирование интегрирование, экстраполирование), графическое отображение результатов.

Теоретическое значение давления $p(y)=\operatorname{arctg}(y+1)-\operatorname{arctg}(y-1)$ было зашумлено. Аппроксимация проводилась при помощи одной обратной квадрики:

```
x = (-4:0.1:4)';
gdata = atan(1*(x+1))+atan(1*(1-x))+0.1*(rand(size(x))-.5);
f = fittype('a*(b^2+(x-c)^2)^(-1)');
gfit = fit(x,gdata,f,'StartPoint',[1 1 1])
```

Результат аппроксимации представлен на рис. 1

```
plot(gfit,x,gdata);
```

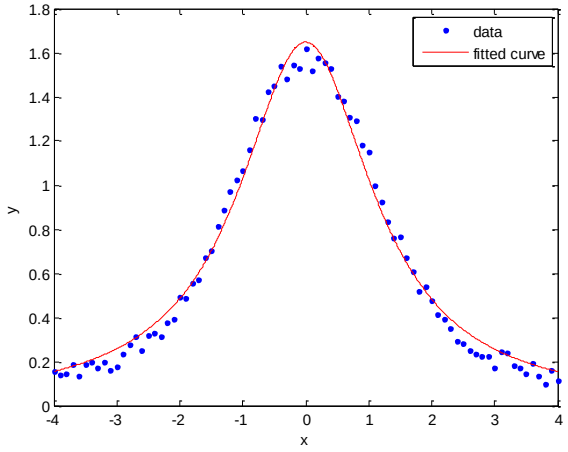


Рис. 1. Сглаживание данных

Значения параметров модели имеют вид

```
gfit =
 General model:
 gfit(x) = a*(b^2+(x-c)^2)^(-1)
```

Coefficients (with 95% confidence bounds):
a = 2.72 (2.595, 2.845)
b = 1.28 (1.243, 1.317)
c = -0.00481 (-0.02981, 0.02019)

На рис. 2 представлена нормальная компонента тензора напряжений σ_y .



Рис. 2. Нормальная компонента σ_y при $x = 1$

Аппроксимация компонент σ_x, τ_{xy} также удовлетворительная.

Библиографический список

1. Уроев, В. М. Уравнения математической физики / В. М. Уроев. – М. : ИФ Яуза, 1998.
2. Снеддон, И. Н. Преобразования Фурье / И. Н. Снеддон. – М. : ИЛ, 1955. – 668 с.
3. Горбаченко, В. И. Решение краевых задач математической физики с помощью сетей радиальных базисных функций / В. И. Горбаченко, М. В. Жуков // Журнал вычислительной математики и математической физики. – 2017. – Т. 57, № 1. – С. 133–143.
4. Горбань, А. Н. Обобщенная аппроксимационная теорема и вычислительные возможности нейронных сетей / А. Н. Горбань // Сибирский журнал вычислительной математики. – 1998. – Т. 1, № 1. – С. 12–24.
5. Овский, А. Г. Моделирование схемы решения трехмерной задачи теории упругости в системе Maple / А. Г. Овский, В. О. Толок // Гідроакустичний журнал. – 2008. – № 3. – С. 88–97.
6. Матросов, А. В. Maple 6. Решение задач высшей математики и механики / Ф. А. Матросов. – СПб. : БХВ-Петербург, 2001. – 528 с.

М. А. Акинтеев

Пензенский государственный университет, Пенза, Россия

К ВОПРОСУ О МАТЕМАТИЧЕСКИХ МЕТОДАХ КОДИРОВКИ ЦВЕТОВЫХ МОДЕЛЕЙ ПРИ ОБРАБОТКЕ ИЗОБРАЖЕНИЙ

Рассматриваются основные цветовые модули, их особенности, взаимосвязь и область применения. Также в статье описываются характеристики восприятия различных цветов зрительной системой человека и робота, определяется, какая цветовая модель наиболее подходит для считывания информации с изображения.

Зрительная система человека способна различать сотни тысяч оттенков цветов и их интенсивность. Таким образом, в изображении, большое количество дополнительной информации может содержаться в цвете. Эта информация, может быть использована для упрощения анализа изображения, например, для идентификации объекта на основе цвета [1].

Три независимые величины используются для описания любого конкретного цвета:

– цветовой тон;

– насыщенность;

– интенсивность.

Цветовой тон определяется доминирующей длиной волны. Длина волны видимых цветов находится между 400 нм (фиолетовый) и 700 нм (красный) на электромагнитном спектре.

Насыщенность определяется чистотой цвета, и зависит от количества белого света, который смешивается с цветом. Чистый тон полностью насыщен, т.е. белого света нет. Цветовой тон и насыщенность вместе определяют цветность для данного цвета.

И, наконец, интенсивность определяется по фактическому количеству света. Более светлый цвет, соответственно более интенсивный [2].

Технологии автоматического распознавания изображения используются в ряде систем компьютерного зрения: биометрическая идентификация, человеко-машинный интерфейс, зрения роботов, компьютерная анимация, системы идентификации и детекционирования в фото-видео камерах и так далее. Задача обнаружения на изображении является первым шагом, предобработкой в процессе решения задачи «более высокого уровня» (например, узнавание лица, распознавание выражения лица и так далее) [3]. Для решения данной задачи необходимо выяснить, сколько и каких фильтров достаточно для анализа цветовой информации. Как показывает практика, для решения этой проблемы достаточно всего трех фильтров (красного – red, зеленого – green и синего – blue цветов) [4]. Таким образом, для распознавания образа наиболее подходящей является цветовая модель RGB, эта модель и другие будут рассмотрены в данной статье.

Цветовые модели.

Цветовые модели представляют собой стандартный способ определения конкретного цвета, с помощью 3D-системы координат и подпространства, которое содержит все составляющие цвета конкретной модели. Любой цвет, который может быть определен с использованием модели будет соответствовать одной точке в подпространстве [5].

Модель RGB.

Модель RGB состоит из трех независимых плоскостей, в каждой из который по одному цвету из основных: красного, зеленого и синего. Для определения конкретного цвета нужно указать количество каждого из основных компонентов, присутствующих в цвете. Шкала серого спектра лежит на прямой, соединяющей черные и белые вершины.

Это аддитивная модель, она подходит для смешивания цветов. Модель RGB используется для цветных мониторов и большинства видеокамер [6].

Модель CMYK.

CMYK (голубой, пурпурный, желтый) – это субтрактивная модель. Она подходит для вычитания цветов. То есть, в то время как модель RGB показывает, то что добавляется к черному, чтобы получить определенный цвет, модель CMY показывает, что вычитается из белого [7]. Таким образом, если поверхность покрыта голубым цветом и освещается белым светом, то красный свет не отражается. Аналогично для пурпурного и зеленого, желтого и синего. Это взаимосвязь между моделями RGB и CMY. Модель CMYK применяется в принтерах и фильтрах [8].

Модель HSI.

Данная цветовая модель задает цветовое пространство в терминах следующих составных компонент: Hue – оттенок цвета (красный, синий, зеленый) Диапазон 0–360° (0–100 % в некоторых Реализациях) Saturation – насыщенность цвета (цветовая чистота) Диапазон от 0 до 100 %. Меньшие значения насыщенности делают цвет серым, в то время как большие значения – более «цветным» Intensity – интенсивность цвета [9]. Для многих людей такой способ задания цвета является интуитивно более понятным, чем RGB или CMYK [10].

Модель YIQ.

В цветовой модели YIQ цвет представляется как 3 компоненты – яркость (Y) и две искусственные цветоразностные (I и Q). Сигнал I называется синфазным, Q – квадратур-

ным. Данная модель является результатом перекодирования RGB для цветного телевидения, и является очень важной моделью для обработки цветных изображений [11].

Компонент яркости (Y) содержит всю информацию, необходимую для черно-белого телевидения, и захватывает наше восприятие относительной яркости отдельных цветов. То мы видим зелеными, как намного легче, чем красный, красный легче, чем синий, обозначается их соответствующих весов 0,587 0,299, 0,114 и в первой строке матрицы преобразования выше. Эти веса должны быть использованы при преобразовании цветного изображения в оттенки серого, если вы хотите, чтобы восприятие яркости остаются теми же [12].

Использование модели YIQ было вынужденной мерой. Психофизиологические исследования выяснили, что разрешающая способность глаза в цвете меньше, чем в яркостной составляющей, и глаз, таким образом, мало чувствителен к цвету мелких деталей. За счет этого при создании совместимой системы цветного телевидения удалось уменьшить полосу частот цветоразностных (не содержащих яркостной информации, в отличие от сигналов основных цветов R, G и B) в три-четыре раза. Чтобы уменьшить заметность помехи от цветоразностных сигналов на черно-белых телевизорах, она должна быть как можно более мелкой, что соответствует большей частоте поднесущей. Но при этом верхняя боковая полоса сигнала цветности подавлялась даже при уменьшении полосы пропускания вчетверо, что при квадратурной модуляции приводило к искажению цветовых оттенков [13].

Дальнейшие исследования установили, что к цветовым переходам разного рода глаз имеет разную чувствительность, что позволило сгруппировать так называемые «теплые» и «холодные» оттенки, и в одной группе уменьшить разрешающую способность еще в три раза. Теперь для передачи одного из сигналов было достаточно полосы всего в 0,5 МГц, при этом верхняя и нижняя боковые полосы передавались без ограничений [14]

Применение полутоновых преобразований цвета в изображениях.

Учитывая все эти различные представления цвета, и, следовательно, цветные изображения, возникает вопрос, какой способ лучше применить для обработки изображения?

Один из возможных способов заключается в применении преобразования в каждой цветовой плоскости в модели RGB, но что именно это значит? Если мы хотим увеличить контраст в темном изображении по гистограмме выравнивания, мы можем просто выровнять каждый цвет самостоятельно? Это приведет к совершенно разным оттенкам цветов в нашем преобразованном изображении. Лучше просто применить трансформацию к компоненту интенсивности в модели HSI или к компоненту яркости в модели YIQ изображения, таким образом оставляя цветность неизменной [15].

Переходы между цветовыми моделями всегда связаны с какими-то потерями цветов из-за несоответствия цветового пространства моделей. В модели CMYK невозможно отобразить очень яркие цвета модели RGB, модель RGB, в свою очередь, не способна передать темные густые оттенки модели CMYK, поскольку природа цвета разная [16]. Потери, вызванные конвертированием, невосполнимы. Так, если конвертировать изображение, созданное в цветовой модели RGB, в модель CMYK, часть цветов модели RGB, которые не может передать CMYK, будут преобразованы с искажениями. И при обратном переходе в RGB эти цвета восстановить не удастся [17].

Библиографический список

1. Foley, J. D. Computer Graphics, Principles and Practice / J. D. Foley, A. van Dam, S. K. Feiner, J. F. Hughes. – Addison-Wesley, Reading, 1990.

2. Gonzales, R. C. Digital Image Processing / R. C. Gonzales, R. E. Woods. – Addison-Wesley, Reading, 1992.

3. Newton, I. Opticks / I. Newton. – 4th Edition. – Dover, New York, 1704/1952.

4. Goldstein, E. B. Sensation and Perception / E. B. Goldstein. – Brooks/Cole, California, 1989.

5. Херн, Д. Компьютерная графика и стандарт OpenGL / Д. Херн, М. П. Бейкер. – М. ; СПб., Киев : Вильямс, 2005.

6. Материалы с сайта. – URL: http://www.cambridgeincolour.com/

7. Порев, В. Компьютерная графика / В. Порев. – СПб. ; М. ; Киев ; Дюссельдорф : БХВ-Петербург, 2002.

8. URL: http://www.remlab.ru/
9. URL: http://www.realcolor.ru/
10. URL: http://www.ukr-print.net/
11. URL: http://www.intuit.ru/
12. URL: http://www.webmascon.com/
13. URL: http://sdb.su/comp-grafika/
14. URL: http://www.sernam.ru/
15. URL: http://www.0x99.ru/
16. URL: http://www.ngpedia.ru/id429044p4.html.
17. URL: http://habrahabr.ru/post/229757/

А. В. Гладилин

Пензенский государственный университет, Пенза, Россия

КОНЦЕПЦИЯ БАЗЫ ЗНАНИЙ С ФУНКЦИЕЙ СЕМАНТИЧЕСКОГО АНАЛИЗА ТЕКСТОВЫХ ДОКУМЕНТОВ

Предложена концепция базы знаний с функцией семантического анализа текста, изложены ключевые идеи, отличающие разработку от аналогов, описаны взаимосвязи между компонентами системы, рассмотрены особенности структуры хранения данных.

В данной работе представлена концепция базы знаний, одной из функций которой может выступать семантический анализ электронных текстовых документов.

Для хранения данных со множеством внутренних связей и ссылок лучше всего приспособлены графовые базы данных [1, с. 142]. В качестве базы данных для представления знаний выбрана мультимодельная (поддерживающая хранение данных в виде графов, документов и «ключ-значение») СУБД ArangoDB. Ее преимуществом по сравнению с такими более популярными решениями, как Neo4j и OrientDB, является возможность записывать в качестве свойств документов числовые идентификаторы.

Главными особенностями рассматриваемой базы знаний являются использование «универсальных понятий» и раздельное хранение морфологии и семантики.

Одна из ключевых особенностей концепции – выделение своеобразных «универсальных понятий», которые одинаковы в представлении любого человека независимо от его родного языка (рис. 1).

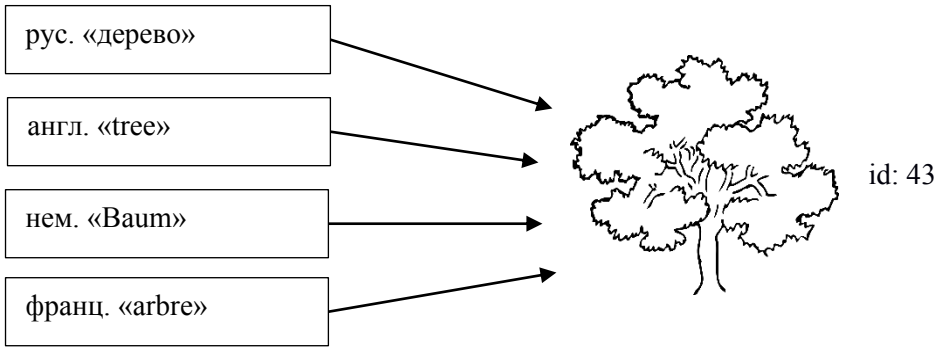


Рис. 1. Пример соответствия слов естественных языков с универсальным понятием

Универсальным понятиям присваивается уникальный числовой идентификатор (например, 432683), и они отражают семантическую часть слова. Многие значения слов естественных языков абстрактны и расплывчаты, что позволяет соединить эквивалентные

значения с помощью численных идентификаторов. Однако для слов с более специфическим значением семантика определяется с помощью специального выражения, составленного из ссылок на другие универсальные понятия.

Использование численных идентификаторов вместо строковых обусловлено тем, что строки могут иметь несколько значений, что может породить нежелательные смысловые коллизии и требует дополнительного уточнения.

Любое слово находит свое выражение на нескольких языковых уровнях: фонетико-фонологическом, морфемно-морфологическом, синтаксическом, семантическом и прагматическом [2, с. 78]. Фонетико-фонологический, морфемно-морфологический и семантический уровни слова (проще говоря, произношение, состав слова и его морфологические свойства) могут быть объединены в одном документе, описывающем слово, синтаксический и прагматический уровни принадлежат предложению и тексту соответственно.

Морфология и семантика разделены по разным базам данных. Каждый из естественных языков (русский, английский, немецкий и т.д.) хранится в отдельной коллекции документов и никаким образом не пересекается с другими языками. Необходимые для морфологического анализа части слов (морфемы – префиксы, корни, суффиксы, окончания) также хранятся в виде отдельных коллекций документов и связаны ссылками, которые описывают морфемный состав слов. Разделение морфологии и семантики необходимо, чтобы выделить одну базу данных, посвященную семантике, к которой потом возможно подключать новые базы данных естественных языков по мере их наполнения.

И содержимое баз данных естественных языков, и базы данных универсальных понятий хранится в формате JSON, часто используемом в современной веб-разработке для построения API [3, с.1]. Особенностью хранения объектов является использование идентификаторов универсальных понятий вместо традиционных строковых идентификаторов в качестве свойств и значений: это нужно для максимальной связности данных между собой. В результате получается граф со множеством вершин и ребер, любая точка которого так или иначе связана со всеми остальными.

Рассмотрим различия между форматом хранения слов на естественном языке и универсальных понятий (рис. 2).

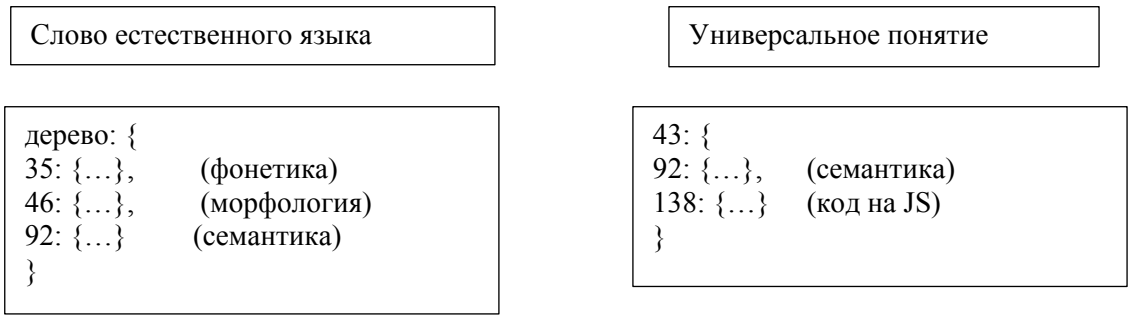


Рис. 2. Связи внутри базы знаний

Для краткости содержимое объектов скрыто. Содержимое объекта, представляющего слово естественного языка, состоит из трех объектов, каждый из которых отображает определенный уровень языка (в фонетике – транскрипция слова, в морфологии – морфемный состав слова и его морфологические свойства, в семантике – массив значений слова со ссылками на универсальные понятия).

Содержимое объекта, представляющего универсальное понятие, состоит из двух частей: первая является семантическим выражением, по структуре напоминающем дефиниции из толкового словаря, вторая часть представлена программным кодом на языке программирования JavaScript, который может выполнен, когда универсальное понятие является каким-либо сложным действием, включающим в себя набор из более простых действий (например, подсчет математической формулы или анализ стиля текста).

Каждому предложению и каждому слову выдается идентификатор соответственно их порядку в тексте. Объект самого верхнего уровня обозначает текст и его свойства (объ-

ект «78» в примере), внутри него содержится объект со всеми предложениями, внутри которых расположены объекты слов и свойства предложения, не отображенные в примере. После проведения семантического анализа объекты слов дополняются синтаксическим и прагматическим уровнем.

Подобный семантический анализ позволяет выделить и проанализировать смысловые связи внутри текста и впоследствии производить запросы к его содержимому, аналогичные вопросам к экспертной системе. Кроме того, выявленные семантические связи открывают возможность осуществлять кластеризацию документов по смыслу предложений или текстов в целом.

Библиографический список

1. Фаулер, М. NoSQL: новая методология разработки нереляционных баз данных : пер. с англ. / Мартин Фаулер, Прамодкумар Дж. Садаладж. – М. : Вильямс, 2013. – 192 с.

2. Гречко, В. А. Теория языкознания : учеб. пособие / В. А. Гречко. – М. : Высш. шк., 2003. – 375 с.

3. Strassner, T. XML vs JSON / T. Strassner // Tufts University. – URL: http://www.cs.tufts.edu/comp/150IDS/final_papers/tstras01.1/FinalReport/FinalReport.html (дата обращения: 27.04.2017).

А. В. Гладилин

Пензенский государственный университет, Пенза, Россия

ОБЗОР СУЩЕСТВУЮЩИХ ПРОГРАММНЫХ КОМПЛЕКСОВ ДЛЯ ПРОВЕДЕНИЯ КЛАСТЕРНОГО АНАЛИЗА

Рассмотрены программные комплексы, включающие в свою функциональность проведение какого-либо вида кластерного анализа, изучены их особенности, перечислены преимущества и недостатки.

Основная цель данной работы – предоставить анализ существующих программных комплексов, в функциональность которых входит какой-либо вид кластерного анализа, выделить их особенности, преимущества и недостатки. Программные комплексы отбираются на основе их популярности и доступности для рядового пользователя.

RStudio – интегрированная среда разработки с открытым исходным кодом для языка программирования R, предназначенная для статистической обработки данных и работы с графикой. Язык программирования R по умолчанию включает в себя минималистичный графический интерфейс, RStudio добавляет следующие возможности: как консоль для выполнения команд, продвинутый редактор кода, средства отладки, браузер объектов, интеграция с документацией, средства для управления графиками [1, с. 2].

Язык программирования R использует менеджер пакетов CRAN, поэтому пользователю предоставляется на выбор около сотни пакетов, предназначенных для работы с кластерным анализом. Среди них представлены методы иерархического (AGNES, CLARA – пакеты cluster, genie, hybridHclust, isopam, protoclust и др.), вероятностного (KMeans, KCentroid, пакеты kmeans, cluster, apcluster, clustMixType и др.), модельно-ориентированного (EM-алгоритм, алгоритмы на основе Байесовской оценки решений) и другие.

Преимущества RStudio: open-source, многообразие пакетов, кроссплатформенность, результат напрямую зависит от навыков пользователя.

Недостатки RStudio: сложность освоения – нужно знать язык программирования R и изучать документацию пакетов.

RapidMiner – программный комплекс, который представляет собой платформу для работы с данными не посредством программирования, а управления блоками, обозначающими ту или иную операцию. Блоки называются «операторами», имеют вход и выход, называемые «портами». На вход принимаются данные, на выход поступают обработанные данные [2, с. 1].

RapidMiner поддерживает следующие виды операторов, осуществляющих кластерный анализ: K-means, X-means, K-medoids, DBSCAN, EM-алгоритм, аггломеративная иерархическая кластеризация, случайная кластеризация.

Преимущества RapidMiner: open-source, кроссплатформенность, удобный и наглядный конструктор операторов.

Недостатки RapidMiner: не поддерживается русский язык.

ClusterEnG – веб-ресурс, предоставляющий возможность осуществить кластерный анализ с графическим отображением результатов. ClusterEnG поддерживает следующие методы кластеризации: K-means, K-medoids, Affinity Propagation, Mixture model, спектральная и иерархическая кластеризация [3, с. 1].

Для того чтобы осуществить кластерный анализ с помощью ClusterEnG, необходимо зайти на сайт проекта, загрузить файл с данными (максимальный объем – 1 Гб), выбрать нужный алгоритм и указать параметры кластеризации (число кластеров или эпсилон), если того требует алгоритм.

Преимущества ClusterEnG: находится в свободном доступе, можно произвести кластерный анализ онлайн.

Недостатки ClusterEnG: данные передаются по сети – при невысокой скорости соединения и большом объеме данных будут сложности, не поддерживается русский язык, небольшие возможности настройки работы алгоритмов.

Weka – программный комплекс, созданный для решения задач машинного обучения (machine learning). Weka использует пакетный менеджер, позволяющий подключать алгоритмы для обработки данных различных категорий (классификация, предобработка, регрессия, кластеризация и другие) на различных языках программирования (R, Python, Clojure) при поддержке популярных технологий (JDBC, Cassandra, Hadoop, Spark).

Осуществление кластерного анализа в Weka происходит путем выбора файла с данными, выбора алгоритма кластеризации и его настройки.

Преимущества Weka: open-source, кроссплатформенность, многообразие пакетов и поддерживаемых технологий.

Недостатки Weka: не поддерживается русский язык.

NCSS – программный комплекс, включающий в свою функциональность около 500 инструментов для работы с графиками и диаграммами и интеллектуального анализа данных. Работа с NCSS очень похожа на работу с редактором электронных таблиц (например, Excel): по умолчанию данные представляются в виде электронной таблицы, затем пользователь выбирает в меню метод обработки данных, настраивает его параметры в появившемся окне, и программа демонстрирует результат в виде отчета.

NCSS поддерживает следующие виды и методы кластеризации: нечеткая, регрессионная, иерархическая аггломеративная кластеризация, K-means, K-medoids.

Преимущества NCSS: огромное количество инструментов для обработки данных, простота освоения.

Недостатки NCSS: платная лицензия (free trial – 30 дней), не поддерживается русский язык.

SPSS Statistics – программный комплекс от компании IBM, схожий по своим возможностям с комплексом NCSS. Данные также представляются в виде электронной таблицы, на выбор предоставляется около сотни инструментов для работы с данными. Кластерный анализ осуществляется аналогично. SPSS поддерживает иерархическую и двухэтапную кластеризацию и метод K-means.

Преимущества SPSS: простота освоения, поддерживается русский язык.

Недостатки SPSS: запутанная документация, платная лицензия (free trial – 14 дней).

Deductor – разработанный в России программный комплекс, предлагающий продвинутую аналитику без программирования, схожий по концепции с RapidMiner (однако

продукты различаются в способе импорта данных). Рабочий процесс в Deductor заключается в импорте файла с данными и последующем выборе и настройки метода обработки данных, результат сразу можно представить в различных способах представления.

Deductor поддерживает методы K-means и G-means, CLOPE, EM-алгоритм, самоорганизующиеся карты Кохонена.

Преимущества Deductor: простота освоения, поддерживается русский язык, поддержка редких методов кластеризации.

Недостатки Deductor: бесплатная версия ограничена в возможностях.

Statistica – программный комплекс от компании Dell, ориентированный на решение задач бизнес-аналитики: анализ, управление, добыча данных. Statistica может быть интегрирована с языками программирования R и Python.

Аналогично SPSS, Statistica поддерживает иерархическую и двухэтапную кластеризацию и метод K-means.

Преимущества STATISTICA: множество поддерживаемых форматов импорта, большое разнообразие инструментов для анализа и визуализации.

Недостатки STATISTICA: не поддерживается русский язык, платная лицензия (free trial – 30 дней).

Orange – программный комплекс, предоставляющий возможность обработки данных с помощью функциональных блоков без применения программирования, что делает его схожим с комплексом RapidMiner, кластерный анализ осуществляется аналогично. Orange поддерживает подключение скриптов на языке Python.

Преимущества Orange: простота освоения, open-source, кроссплатформенность, удобный и простой конструктор операторов.

Недостатки Orange: не поддерживается русский язык.

Подводя итоги, можно отметить такие программные комплексы, как RapidMiner, Weka и Orange, по большинству параметров не уступающие дорогостоящим бизнес-решениям в лице NCSS и SPSS. Более продвинутым пользователям, обладающим минимальным опытом программирования, можно посоветовать изучить язык программирования R и использовать RStudio для более полного контроля процесса кластерного анализа.

Библиографический список

1. Verzani, J. Getting Started with RStudio / J. Verzan. – Sebastopol, O'Reilly, 2011. – 92 p.
2. Крахалев, В. Введение в RapidMiner / В. Крахалев. – URL: habrahabr.ru (дата обращения: 27.04.2017).
3. Manjunath, M. ClusterEnG: An interactive educational web resource for clustering big data / M. Manjunath, Y. Zhang. – URL: http://biorxiv.org/content/biorxiv/early/2017 /03/27/120915.full.pdf (дата обращения: 27.04.2017).

К. Е. Савенков, В. А. Николаева

Пензенский государственный университет, Пенза, Россия

ОБУЧЕНИЕ СЕТЕЙ РАДИАЛЬНЫХ БАЗИСНЫХ ФУНКЦИЙ ПРИ РЕШЕНИИ ЗАДАЧ АППРОКСИМАЦИИ

Были исследованы различные варианты метода Нестерова и метод Левенберга – Марквардта, рассмотрены их плюсы и минусы, сделаны выводы о целесообразности применения каждого из них.

Искусственные нейронные сети являются достаточно мощным инструментом в решении различных нелинейных задач. В частности, сети радиальных базисных (также

встречается радиально-базисных) функций, описываемые в работе, активно используются для решения задач приближения функций [1–3]. Искусственные нейронные сети имеют определенные проблемы, при использовании их на практике. Одна из таких проблем была исследована и описана в текущей работе. Были исследованы между собой метод 1-го порядка (метод Нестерова) и метод 2-го порядка (метод Левенберга-Марквардта). Далее по тексту, под СРБФ будет пониматься сеть радиальных базисных функций, а РБФ будет означать радиальную базисную функцию.

СРБФ – искусственная нейронная сеть, которая использует РБФ как функции активации. Графическая структура такой сети представлена на рис. 1.

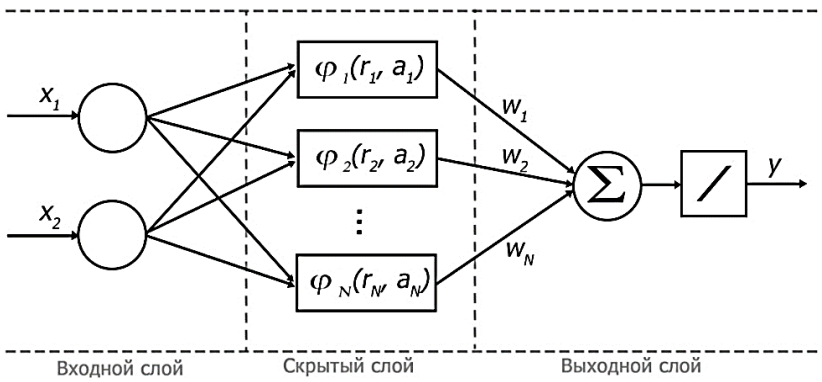


Рис. 1. Структура сети радиальных базисных функций

Где r – Евклидова метрика, задающая расстояние между точкой $\mathbf{x}$ и центром РБФ $\mathbf{c}$. Данная нейронная сеть имеет 3 слоя нейронов [1]. Первый слой осуществляет распределение входных значений между нейронами скрытого слоя. Второй слой совершает нелинейное преобразование значений, полученных с входного слоя нейронов. Третий слой суммирует полученные значения из скрытого слоя нейронов с весами и подает значение на выход сети.

СРБФ осуществляет отображение вида $F:\mathbf{x}\to y$, где $\mathbf{x}=\left[x_1,x_2,...,x_n\right]$. Выход сети вычисляется по формуле:

$$f\left(\mathbf{x}\right)=\sum_{i=1}^{N}w_i\varphi\left(\left\|\mathbf{x}-\mathbf{c}_i\right\|,a_i\right),$$

где N – количество нейронов скрытого слоя, $\mathbf{w}=\left[w_1,w_2,...,w_N\right]$ – веса линейного сумматора. В качестве РБФ использовалась Гауссова функция

$$\varphi\left(\left\|\mathbf{x}-\mathbf{c}_i\right\|,a_i\right)=\exp\left(-\frac{\left\|\mathbf{x}-\mathbf{c}_i\right\|^2}{2a_i^{\,2}}\right),$$

где $\mathbf{a}=\left[a_1,a_2,...,a_N\right]$ является параметром ширины радиальных базисных функций. Выражение $\left\|\mathbf{x}-\mathbf{c}_i\right\|$ вычисляет расстояние между точкой $\mathbf{x}$ и центром $\mathbf{c}_i$ РБФ

$$\left\|\mathbf{x}-\mathbf{c}_i\right\|=\sqrt{\left(x_1-c_{1i}\right)-\left(x_2-c_{2i}\right)-...-\left(x_n-c_{ni}\right)}\,.$$

Пусть $f(x)$ – функция, задающая отображение входных значений СРБФ в выходные, т.е. $f(x)$ и есть СРБФ. Тогда, при обучении сети, будем пытаться минимизировать функционал среднеквадратичной ошибки

$$MSE(\mathbf{o},\mathbf{t}) = \frac{1}{n}\sum_{i=1}^{n}\left(o_i - t_i\right)^2,$$

где $\mathbf{o} = \left[o_1, o_2, ..., o_n\right]$ являются выходами, а $\mathbf{t} = \left[t_1, t_2, ..., t_n\right]$ целевыми значениями $f(x)$, n равно количеству объектов обучающей выборки. Существует множество оптимизационных алгоритмов, мы рассмотрим **метод Нестерова**, из методов 1-го порядка, и **метод Левенберга-Марквардта**, являющимся методом 2-го порядка.

В обычном методе градиентного спуска на каждом шаге итерации следующее приближение получается из предыдущего вычитанием вектора градиента, умноженного на шаг η_k

$$\boldsymbol{\theta}^{(k)} = \boldsymbol{\theta}^{(k-1)} - \eta^{(k)}\nabla_{\theta}Q\left(\boldsymbol{\theta}^{(k-1)}, \mathbf{X}\right),$$

где $Q\left(\boldsymbol{\theta}^{(k-1)}, \mathbf{X}\right)$ – функционал ошибки алгоритма. При этом выражение для градиента в матричной форме имеет вид

$$\nabla_{\theta}Q\left(\boldsymbol{\theta}, \mathbf{X}\right) = \frac{2}{n}\mathbf{X}^T\left(\mathbf{X}\boldsymbol{\theta} - y\right).$$

Выражение для j-й компоненты градиента, таким образом, содержит суммирование по всем объектам обучающей выборки

$$\frac{\partial Q}{\partial \theta_j} = \frac{2}{n}\sum_{i=1}^{n} x_i^j\left(\left\langle \boldsymbol{\theta}, x_i\right\rangle - y_i\right).$$

В этом и состоит основной недостаток метода градиентного спуска – в случае большой выборки даже одна итерация метода градиентного спуска будет производиться долго.

Идея стохастического градиентного спуска основана на том, что в сумме в выражении для j-й компоненты градиента i-е слагаемое указывает то, как нужно поменять параметр θ_j, чтобы качество увеличилось для i-го объекта выборки. Вся сумма при этом задает, как нужно изменить этот вес, чтобы повысить качество для всех объектов выборки. В стохастическом методе градиентного спуска градиент функционала качества вычисляется только на одном случайно выбранном объекте обучающей выборки. Это позволяет обойти вышеупомянутый недостаток обычного градиентного спуска. Таким образом, алгоритм стохастического градиентного спуска следующий. Сначала выбирается начальное приближение, например, $\mathbf{w}^0 = (0,0,...,0)$. Далее, последовательно вычисляются итерации $\boldsymbol{\theta}^{(k)}$: сначала случайным образом выбирается объект $\mathbf{x}_i$ из обучающей выборки $\mathbf{X}$ и вычисляется вектор градиента функционала качества на этом объекте, а следующее приближение получается из предыдущего вычитанием умноженного на шаг η_{θ} полученного вектора

$$\boldsymbol{\theta}^{(k)} = \boldsymbol{\theta}^{(k-1)} - \eta_{\theta}\nabla_{\theta}Q(\boldsymbol{\theta}^{(k-1)}, \mathbf{x}_i).$$

Итерации прекращаются при достижении определенного условия, например,

$$\left\|\boldsymbol{\theta}^{(k)} - \boldsymbol{\theta}^{(k-1)}\right\| < \varepsilon.$$

Между этими двумя видами градиентного спуска существует компромисс, называемый иногда «**mini-batch**». В этом случае обновления происходят по некоторому числу обучающих примеров. Это, во-первых, ускоряет вычисление градиента, во-вторых позволяет (теоретически) избежать локальных минимумов.

Основная идея **метода Нестерова** заключается в том, чтобы осуществить оптимизацию, используя метод спуска с моментом [4, 5]. Изменения параметров происходят по формуле $\boldsymbol{\theta}^{(k)} = \boldsymbol{\theta}^{(k-1)} - \mathbf{v}^{(k)}$, где $\mathbf{v}^{(k)}$ – вектор изменения параметров сети (скорость движения). Однако вычисляется градиент по уже измененным параметрам, а не исходным. Формула имеет вид:

$$\mathbf{v}^{(k)} = \gamma \mathbf{v}^{(k-1)} + \eta_\theta \nabla_\theta Q\left(\boldsymbol{\theta}^{(k-1)} - \gamma \mathbf{v}^{(k-1)}\right),$$

где γ – коэффициент сохранения скорости. Физическая интерпретация заключается в том, что объект, катящийся по горизонтальной поверхности будет еще некоторое время сохранять свою скорость, что и отражает этот метод. Вдобавок к этому, мы вычисляем значения сразу в следующей точке. Очевидные плюсы в том, что возможно «перепрыгивать» некоторые локальные минимумы.

Применительно к нашей структуре, для изменения параметров будем вычислять градиенты по каждому из параметров сети. Опуская несложные, но громоздкие вычисления, запишем формулы вычисления компонентов градиента функционала сразу. Компонент градиента функционала по весу вычисляется следующим образом:

$$\frac{\partial Q^{(k)}}{\partial w_i^{(k)}} = \sum_{j=1}^{N}\left(f\left(\mathbf{p}_j\right) - t_j \right)\varphi_i\left(\mathbf{p}_j\right).$$

Компонент градиента функционала по центрам

$$\frac{\partial Q^{(k)}}{\partial c_{i1}^{(k)}} = w_i \sum_{j=1}^{N}\left(f\left(p_{j1}\right) - t_j \right)\varphi_i\left(\mathbf{p}_j\right)\frac{p_{j1} - c_{i1}}{a_i^2}.$$

Аналогично,

$$\frac{\partial Q^{(k)}}{\partial c_{i2}^{(k)}} = w_i \sum_{j=1}^{N}\left(f\left(p_{j2}\right) - t_j \right)\varphi_i\left(\mathbf{p}_j\right)\frac{p_{j2} - c_{i2}}{a_i^2}.$$

Компонент градиента функционала по ширине

$$\frac{\partial Q^{(k)}}{\partial a_i^{(k)}} = w_i \sum_{j=1}^{N}\left(f\left(\mathbf{p}_j\right) - t_j \right)\varphi_i\left(\mathbf{p}_j\right)\frac{\left\|\mathbf{p}_j - \mathbf{c}_i\right\|^2}{a_i^3}.$$

Используя вычисленные градиенты функционала корректируются веса

$$\mathbf{w}^{(k)} = \mathbf{w}^{(k-1)} - \left(\gamma_w \mathbf{w}^{(k-1)} + \eta_w \frac{\partial Q^{(k)}}{\partial w^{(k-1)}} \right),$$

центры

$$\mathbf{c}^{(k)} = \mathbf{c}^{(k-1)} - \left(\gamma_c \mathbf{c}^{(k-1)} + \eta_c \frac{\partial Q^{(k)}}{\partial c^{(k-1)}} \right),$$

и ширина РБФ

$$\mathbf{a}^{(k)} = \mathbf{a}^{(k-1)} - \left(\gamma_a \mathbf{a}^{(k-1)} + \eta_a \frac{\partial Q^{(k)}}{\partial a^{(k-1)}} \right).$$

Каждый вектор параметров, в вычислении градиента, должен участвовать в виде $\boldsymbol{\theta} - \gamma_\theta \mathbf{v}^{(k-1)}$, где $\boldsymbol{\theta}$ – текущий вектор параметров, γ_θ – коэффициент сохранения скорости, $\mathbf{v}^{(k-1)}$ – вектор значений на прошлой итерации.

Метод Левенберга – Марквардта старается найти вектор оптимальных параметров функции из решения СЛАУ (системы линейных алгебраических уравнений) [3]

$$\left(\mathbf{J}_{(k)}^{\mathrm{T}}\mathbf{J}_{(k)}+\mu_{(k)}\mathbf{E}\right)\Delta\boldsymbol{\theta}_{(k+1)}=-\mathbf{g}_{(k)},$$

где $\Delta\boldsymbol{\theta}_{(k)}$ есть вектор смещения. Изменение параметров, используя смещение $\Delta\boldsymbol{\theta}_{(k)}$ осуществляется по формуле $\boldsymbol{\theta}_{(k)}=\boldsymbol{\theta}_{(k-1)}+\Delta\boldsymbol{\theta}_{(k)}$. $\mathbf{E}$ – единичная матрица; $\mu_{(k)}$ – параметр регуляризации, изменяющийся на каждой эпохе обучения; $-\mathbf{g}_{(k)}$ – вектор градиент, может быть приближен следующим произведением $\mathbf{g}_{(k)}=\mathbf{J}^{\mathrm{T}}\mathbf{e}$, где $\mathbf{e}$ – вектор ошибок. Матрица Якоби $\mathbf{J}_{(k)}$, вычисленная по параметрам сети на k -й эпохе обучения в общем случае имеет вид

$$\mathbf{J}=\begin{bmatrix}
\frac{\partial e_1}{\partial w_1} & \cdots & \frac{\partial e_1}{\partial w_N} & \frac{\partial e_1}{\partial c_{11}} & \cdots & \frac{\partial e_1}{\partial c_{N1}} & \frac{\partial e_1}{\partial c_{12}} & \cdots & \frac{\partial e_1}{\partial c_{N2}} & \frac{\partial e_1}{\partial a_1} & \cdots & \frac{\partial e_1}{\partial a_N} \\
\frac{\partial e_2}{\partial w_1} & \cdots & \frac{\partial e_2}{\partial w_N} & \frac{\partial e_2}{\partial c_{11}} & \cdots & \frac{\partial e_2}{\partial c_{N1}} & \frac{\partial e_2}{\partial c_{12}} & \cdots & \frac{\partial e_2}{\partial c_{N2}} & \frac{\partial e_2}{\partial a_1} & \cdots & \frac{\partial e_2}{\partial a_N} \\
\vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
\frac{\partial e_n}{\partial w_1} & \cdots & \frac{\partial e_n}{\partial w_N} & \frac{\partial e_n}{\partial c_{11}} & \cdots & \frac{\partial e_n}{\partial c_{N1}} & \frac{\partial e_n}{\partial c_{12}} & \cdots & \frac{\partial e_n}{\partial c_{N2}} & \frac{\partial e_n}{\partial a_1} & \cdots & \frac{\partial e_n}{\partial a_N}
\end{bmatrix}.$$

Для изменения параметров будем вычислять матрицы Якоби по каждому из параметров сети. Опуская несложные, но громоздкие вычисления, запишем формулы вычисления компонента матрицы Якоби сразу. Компонент матрицы функционала вычисляется следующим образом

$$\frac{\partial e_i}{\partial w_j}=\varphi_j\left(\mathbf{p}_i\right).$$

Компонент матрицы по центрам

$$\frac{\partial e_i}{\partial c_{j1}}=w_j\varphi_j\left(\mathbf{p}_i\right)\frac{p_{i1}-c_{j1}}{a_j^2}.$$

Аналогично,

$$\frac{\partial e_i}{\partial c_{j2}}=w_j\varphi_j\left(\mathbf{p}_i\right)\frac{p_{i2}-c_{j2}}{a_j^2}.$$

Компонент матрицы по ширине

$$\frac{\partial e_i}{\partial a_j}=w_j\varphi_j\left(\mathbf{p}_i\right)\frac{\left\|\mathbf{p}_i-\mathbf{c}_j\right\|^2}{a_j^3}.$$

Используя вычисленные компоненты матрицы Якоби, корректируются веса, исходя из решения СЛАУ

$$\left(\mathbf{J}_{(k)}^{\mathrm{T}}\mathbf{J}_{(k)}+\mu_{(k)}\mathbf{E}\right)\Delta\boldsymbol{\theta}_{(k+1)}=-\mathbf{g}_{(k)}$$

Веса изменяются по формуле $\mathbf{w}_{(k)}=\mathbf{w}_{(k-1)}+\Delta\boldsymbol{\omega}_{(k)}^{w}$, центры $\mathbf{c}_{(k)}=\mathbf{c}_{(k-1)}+\Delta\boldsymbol{\omega}_{(k)}^{c}$ и ширина РБФ $\mathbf{a}_{(k)}=\mathbf{a}_{(k-1)}+\Delta\boldsymbol{\omega}_{(k)}^{a}$. Векторы $\Delta\boldsymbol{\omega}_{(k)}^{w}$, $\Delta\boldsymbol{\omega}_{(k)}^{c}$ и $\Delta\boldsymbol{\omega}_{(k)}^{a}$ являются срезами вектора $\Delta\boldsymbol{\omega}_{(k)}$ по каждому из параметров.

В ходе исследовательской работы решалась задача приближения функции

$$f(x,y)=x^2+y^2, x,y\in\left[-3;3\right].$$

График приближаемой функции представлен на рис. 2.

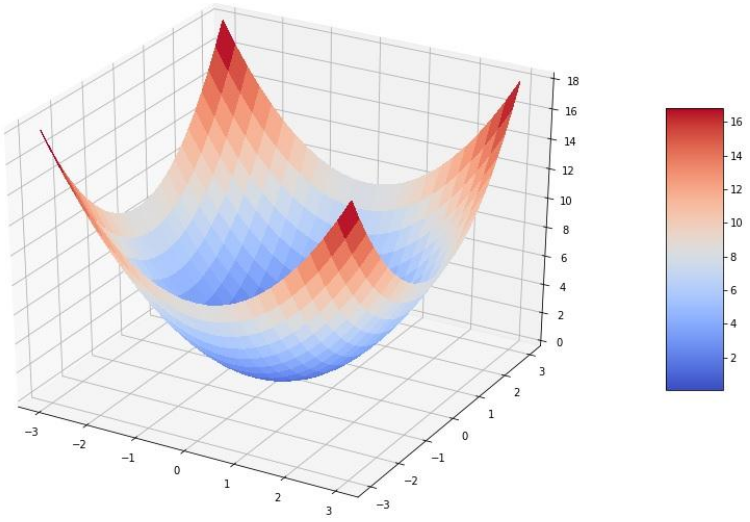


Рис. 2. График приближаемой функции

Задача решалась, используя 3 вариации метода Нестерова (batch, mini-batch, stochastic) и метод Левенберга-Марквардта. Обучение осуществлялось на одинаковых, для всех методов, начальных параметрах СРБФ. Графики сходимости каждого алгоритма были выведены на отдельный график и представлены на рис. 3.

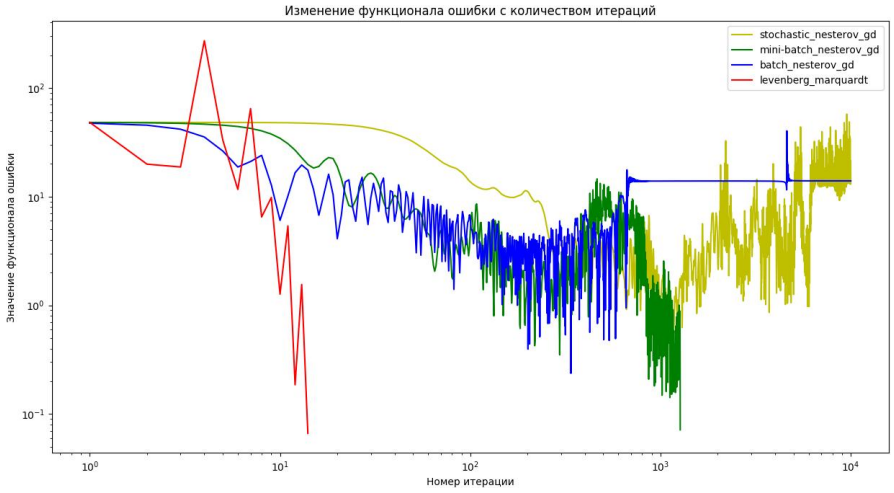


Рис. 3. График изменения функционала ошибки

Результаты тестирования алгоритмов сведены в таблицу ниже.

Таблица

Результаты тестирования алгоритмов

Алгоритм	Время обучения	Кол-во итераций	Достигнутая ошибка
batch_nesterov	1h 11min 55sec	9999	13,9648360012
mini-batch_nesterov	1min 32 sec	680	0,0793991705652
stochastic_nesterov	10min 55sec	9999	13,1386113913
levenberg-marquardt	6 sec 17 ms	13	0,066295492763

Обучение осуществлялось на машине с OS: Ubuntu 16.04 xenial, RAM: 4GB, CPU: Intel Pentium CPU B940 @ 2GHz.

Из таблицы видно, что хорошие результаты были достигнуты лишь двумя алгоритмами, это mini-batch_nesterov и levenberg-marquardt. Чтобы решить проблему останова методов было решено использовать 10000 итераций, как предельное количество, также обучение останавливалось, если алгоритм достигал значения среднеквадратичной ошибки в 0,1. Отметим плюсы и минусы в реализации и обучении алгоритмов-«победителей». Они оба нуждаются в тщательной настройке параметров, что требует некоторого терпения и времени. Также существует проблема выбора начальных параметров. Метод Нестерова намного чувствительнее к начальному приближению, в отличие от метода Левенберга-Марквардта. Однако стоит отметить, что метод Нестерова гораздо проще в реализации, нежели метод 2-го порядка. Алгоритм Левенберга-Марквардта требует решения СЛАУ, что в некоторых ситуациях трудно реализуемо из-за особого вида матрицы.

Авторы выражают благодарность доктору технических наук, профессору Горбаченко Владимиру Ивановичу за оказанную помощь и направление при написании данной работы.

Библиографический список

1. Kriesel, D. A Brief Introduction to Neural Networks / D. Kriesel. – 2007. – URL: www.dkriesel.com

2. Хайкин, С. Нейронные сети: полный курс / С. Хайкин. – М. : Вильямс, 2006. – 1104 с.

3. Осовский, С. Нейронные сети для обработки информации / С. Осовский. – М. : Горячая линия-Телеком, 2016. – 448 с.

4. Ruder, S. An overview of gradient descent optimization algorithms / S. Ruder. – 2017. – URL: http://ruder.io/optimizing-gradient-descent/index.html#gradientdescentvariants

5. Goodfellow, I. Deep Learning (Adaptive Computation and Machine Learning series) / I. Goodfellow, Y. Bengio, A. Courville. – 2016. – URL: http://www.deeplearningbook.org/

А. С. Никонов, И. П. Бурукина

Пензенский государственный университет, Пенза, Россия

РАЗРАБОТКА КОМПОНЕНТОВ СИСТЕМЫ WAVELET-ПРЕОБРАЗОВАНИЯ ИЗОБРАЖЕНИЯ

Рассмотрена теория wavelet-преобразования, ее применение к удалению шума на изображении. Рассмотрены аналогичные методы: линейный, медианный, адаптивный фильтры. Проведено сравнение результатов.

Представление информации в графическом виде является неотъемлемой частью деятельности человека. Графическая информация применялась еще в первобытном обществе в виде наскальной живописи. В настоящее время, люди используют графику в различных областях науки, культуры, искусства, медицины, а так же в инженерной деятельности, военном деле, в логистике, кибернетике и т.д.

Работа с изображениями связана с определенными трудностями. В процессе обработки, передачи, хранения и, возможно, создания оно может быть повреждено или искажено. На изображении могут появиться шумы. Это приводит к ухудшению восприятия информации человеком.

Рассмотрим следующие виды шума:

- соль и перец (случайные пиксели становятся черными или белыми);
- гауссов шум (колебания яркости пикселей, определяемые нормальным законом);
- мультипликативный шум (пиксели изображения умножаются на случайное значение).

Для решения задачи очистки изображения от шума создаются программные средства на основе определенных математических или алгоритмических методов.

Существуют различные методы удаления шума на изображении. В общем случае их можно разделить на точные и эвристические. К эвристическим методам можно отнести различные фильтры, например, линейный, медианный и адаптивный. Преобразование Фурье и wavelet-преобразование являются математическими способами решения проблемы.

Преобразование Фурье анализирует исходную информацию, например, сигнал, тригонометрическими функциями. Математический аппарат основывается на разложении функции в ряд. Преобразование оперирует комплексными числами.

К недостаткам преобразования Фурье можно отнести требование знания сигнала не только в прошлом, но и в будущем [3], невозможность точного восстановления сигнала, трудности, связанные с вычислением комплексных чисел, не удовлетворительные результаты работы с сигналами, размерностью более двух.

В эвристических фильтрах существует понятие апертуры (или окна). Это квадратная матрица небольшого размера, внутри которой происходит обработка пикселов исходного изображения.

Линейный фильтр включает в себя усредняющий фильтр и фильтр Гаусса. Работа линейного фильтра основана на усреднении значений внутри окна. Фильтрация заключается в умножении ядра на элемент матрицы исходного изображения. В результате получается очищенное изображение.

Во многих случаях линейным фильтром создается рамка вокруг изображения. Помимо этого, использование линейного фильтра приводит к сглаживанию изображения. Снижается резкость, и изображение «размывается». Но при небольшой апертуре фильтр может показать допустимый результат. В целом результаты работы фильтра не являются высокими. В настоящее время к результатам решения подобных задач выдвигаются более высокие требования.

Работа медианного фильтра заключается в вычислении медианного значения пикселей. Медианный фильтр, в отличие от линейного, не создает рамок на изображении и не размывает его.

Адаптивный фильтр применяется для устранения гауссовского шума. Медианный фильтр показывает хороший результат при невысокой степени зашумленности [2]. Было выявлено, что данный фильтр применим и для сильно зашумленных изображений.

Адаптивный фильтр не приводит к появлению рамок, а так же изображение не размывается.

Таким образом, медианный и адаптивный фильтры показывают более высокий результат, чем линейный. Фильтры могут быть применены в определенных задачах.

Достоинством эвристических методов является простота реализации по сравнению с wavelet-преобразованием. Разработка программы, использующей wavelet-декомпозицию, является более трудоемким процессом, но приносит более высокие результаты.

Wavelet-преобразование позволяет устранить шум или, в некоторых случаях, снизить зашумленность изображения, не зависимо от вида шума. Это является преимуществом по сравнению с ранее рассмотренными методами.

В начале XX в. венгерский математик Альфред Хаар определил понятие вейвлета. Разработкой теории wavelet-преобразования так же занимались Добеши, Морле, Маллат.

Понятие вейвлета определяется, как математическая функция в виде короткой волны (всплеска), применяемая для обработки сигналов или изображений. Wavelet в переводе с английского означает «короткая волна».

Применение вейвлетов выходит за границы работы с изображениями. Вейвлеты применяются в задачах распознавания образов, в радиотехнике. Важной сферой применения wavelet-анализа является медицина. В этой области вейвлеты можно использовать для анализа ЭКГ. Помимо этого, wavelet-преобразование используется для сжатия информации (например, сигналов, изображений и т.д.).

Математическим аппаратом, использующим вейвлеты, называется wavelet-преобразование. Wavelet-преобразование представляет собой скалярное произведение вейвлета и анализируемого сигнала (например, изображения)[1].

Wavelet-декомпозиция позволяет разделить рассматриваемый сигнал или изображение на составляющие. Это осуществляется при помощи вейвлетов.

Дискретное wavelet-преобразование позволяет получить аппроксимирующие и детализирующие коэффициенты, которые описывают сглаженный сигнал и его колебания соответственно[5].

Алгоритм многоуровневого wavelet-преобразования называется алгоритмом Малла-та. Алгоритм позволяет получить аппроксимирующие и детализирующие коэффициенты. Они определяются при помощи высокочастотного фильтра для детализирующих коэффициентов и низкочастотного фильтра для аппроксимирующих коэффициентов. Затем выполняется «прореживании» сигнала (децимация). На следующей итерации алгоритма аппроксимирующие коэффициенты подвергаются аналогичному воздействию.

Как было сказано ранее, wavelet-преобразование позволяет получить аппроксимирующие и детализирующие коэффициенты. Аппроксимирующая часть содержит существенную информацию об изображении, а детализирующая – детали изображения. Следовательно, шумовая составляющая имеется в детализирующих коэффициентах, значит, для устранения шума их можно отбросить[4].

Вторым способом очистки изображения является трешолдинг (от англ. thresholding). Предполагается, что шумовая составляющая является значением, меньшим по модулю, чем основная информация. Это означает, что для очистки изображения можно сделать нулевыми коэффициенты, значение которых меньше определенного значения (порога).

Второй способ является более трудоемким. Его реализация приводит к применению определенной декомпозиции и дальнейшего восстановления информации.

Важно отметить, что wavelet-преобразование обладает свойством точного восстановления сигнала. Для этого базовый wavelet должен быть ортогональным.

Таким образом, определив базовый wavelet, уровень декомпозиции, значение порога обработки и применив рассматриваемый аппарат, появляется возможность фильтрации изображения. Результат не зависит от вида шума.

На рис. 1 представлены результаты очистки изображения описанными ранее способами.

Исходное изображение	Зашумленное изображение
	
Wavelet-фильтрация	Усредняющий фильтр

Рис. 1. Результаты фильтрации изображения различными методами (начало)





Фильтр Гаусса

Медианный фильтр





Рис. 1. Результаты фильтрации изображения различными методами (окончание)

Анализируя увиденное, можно сделать вывод, что wavelet-фильтрация очистила изображение, показав хороший результат. Остатков шума не было замечено. В некоторых местах имеется незначительное размытие. При этом в очищенном изображении цвета похожи на исходные, и само изображение похоже на исходное.

Библиографический список

1. Астафьева, Н. М. Вейвлет-анализ: основы теории и примеры применения / Н. М. Астафьева // Успехи физических наук. – 2014. – № 11. – С. 1145–1170.

2. Бондина, Н. Н. Сравнительный анализ алгоритмов фильтрации медицинских изображений / Н. Н. Бондина, А. С. Калмычков, В. Э. Кривенцов // Вестник Национального технического университета. Харьковский политехнический институт. Сер.: Информатика и моделирование. – 2012. – № 38. – С. 14–26.

3. Дьяконов, В. П. Matlab. Обработка сигналов и изображений : специальный справочник / В. П. Дьяконов, И. В. Абраменкова. – СПб. : Питер, 2012. – 608 с.

4. Московский, С. Б. Очистка сигнала от шумов с использованием вейвлет-преобразования / С. Б. Московский, А. Н. Сергеев, Н. А. Лалина // Universum: Технические науки : электрон. науч. журн. – 2015. – № 2. – URL: http://7universum.com/ru/tech/archive/item/1958 (дата обращения: 27.02.2017).

5. Смоленцев, Н. К. Основы теории вейвлетов. Вейвлеты в Matlab / Н. К. Смоленцев. – 2-е изд., перераб. и доп. – М. : Изд-во ДМК, 2015. – 304 с.

Д. В. Горбачев

Оренбургский государственный университет, Оренбург, Россия

МОДЕЛИРОВАНИЕ ПРОЦЕССА ИНСТРУМЕНТАЛЬНОЙ ДИАГНОСТИКИ ЗАБОЛЕВАНИЯ СИСТЕМОЙ МАССОВОГО ОБСЛУЖИВАНИЯ И ОТНОСИТЕЛЬНЫМИ ПРИОРИТЕТАМИ

Для современного медицинского учреждения важным и актуальным является вопрос обеспечения высокого качества медицинской помощи. Одним из аспектов качества медицинской помощи является организация процессов инструментальной диагностики заболеваний. В статье предлагается подход к разработке модели оценки характеристик подсистемы инструментальной диагностики заболеваний в целях эффективного управления данным процессом.

Одной из важнейших задач врача является ранняя постановка правильного диагноза, позволяющего своевременно начать соответствующее лечение. Стандартами оказания медицинской помощи устанавливаются следующие виды медицинской диагностики:

- прием (осмотр, консультация) врача-специалиста;
- лабораторные методы исследования;
- инструментальные методы исследования.

В общем случае процесс диагностики заболевания реализуется следующим образом.

В кабинете врача диагноз устанавливается по данным анамнеза приблизительно более чем в 50 % случаев, на основании объективного клинического исследования – приблизительно в 30 % и по лабораторным данным – приблизительно в 20 % [1].

Диагностический процесс можно подразделить на три взаимосвязанных этапа: постановка первичного диагноза (предварительная гипотеза), построение дифференциально-диагностического ряда (выдвижение дополнительных гипотез), окончательный диагноз (обоснование окончательной гипотезы) [2].

По завершению этапа диагностики больному назначается лечение вид, форма и продолжительность которого определяется поставленным диагнозом и соответствующим стандартом оказания медицинской помощи.

Процесс диагностики заболевания с использованием медицинского оборудования формализуется системой массового обслуживания (рис. 1).

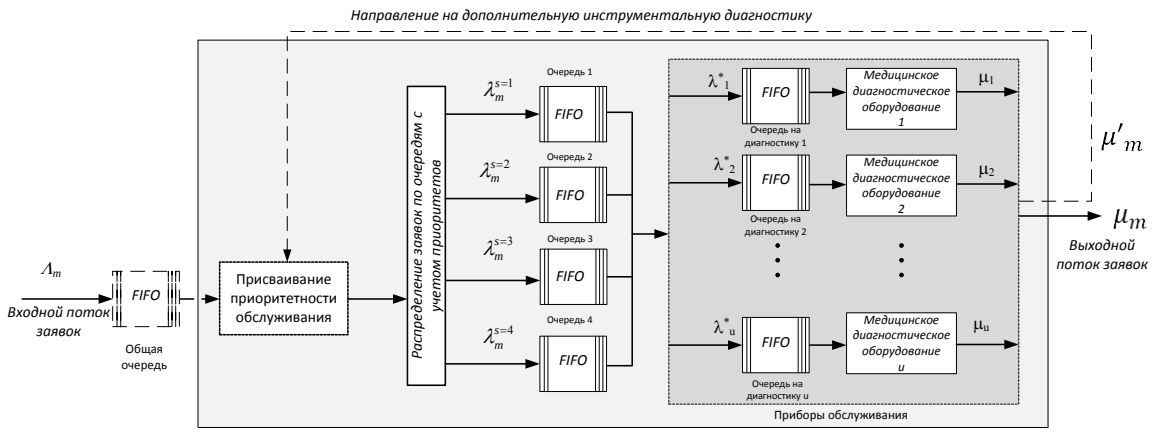


Рис. 1. Структура СМО инструментальной диагностики заболевания

Функционирование СМО происходит исходя из учета следующих особенностей системы и вводимых ограничений:

- в качестве входного потока рассматривается поток пациентов по одной нозологической форме, у которых может быть разная степень тяжести заболевания;
- система является многоканальной с ограничением по длине очередей. Размер каждой очереди ограничивается исходя из условия максимально возможного обслуживания пациентов в течение рабочего времени медицинского прибора;
- общий входной поток и потоки, формирующие очереди к приборам, рассматриваются как простейшие, поскольку вероятностные характеристики потока пациентов не зависят от времени, т.е. поток стационарен, число обращений пациентов в любом интервале времени является независимым от времени обслуживания, т.е. поток без последействия, вероятность одновременного поступления на диагностику двух или более пациентов очень мала по сравнению с вероятностью обращения одного пациента, т.е. поток ординарен;
- система является СМО с относительным приоритетом, в которой заявка высокого приоритета ожидает завершения обслуживания очередной заявки и только после этого поступает на обслуживание. Заявки с одним приоритетом формируют собственные очереди;
- выходной поток системы представляет собой сумму потоков продиагностированных пациентов, диагноз которых установлен, и пациентов, направленных на дополнительное исследование.

На вход системы поступает поток пациентов (больных) с интенсивностью Λm, который формирует общую очередь размером N. Во время первичного осмотра врачом для пациента устанавливается степень тяжести заболевания $s = 1…4$ (табл. 1), в зависимости от чего в модели определяется приоритет обслуживания (диагностики) пациента. Все пациенты, обладающие одинаковым приоритетом, формируют соответствующие очереди с интенсивностями потоков λ_m^s, где m – нозологическая форма заболевания. Диагностика состояния пациента, как правило, требует проведения исследований с помощью нескольких приборов, поэтому на входе каждого из них может сформироваться своя очередь пациентов на диагностику с интенсивностью входного потока λ_i^*, $i = 1,…,u$. Обслуживание (медицинская диагностика) пациентов в каждом из рассмотренных случаев (т.е. во всех очередях) осуществляется согласно правилу FIFO.

Таблица 1

Приоритеты доступа к медицинскому оборудованию при инструментальной диагностике

Степень тяжести заболевания	Значение приоритета обслуживания
Легкое заболевание, $s_1 = 1$	0,1
Заболевание средней тяжести, $s_2 = 3$	0,2
Тяжелое заболевание, $s_3 = 5$	0,3
Крайне тяжелое заболевание, $s_4 = 7$	0,4

В результате работы системы формируется выходной поток $\mu = \mu_m + \mu'_m$, где μ_m – пациенты прошедшие диагностику и покинувшие систему, μ'_m – пациенты, которым необходимо пройти дополнительные исследования. Задача моделирования заключается в определении характеристик потока μ_m.

Рассматриваемый вариант построения системы массового обслуживания представляет собой параллельно функционирующие подсистемы, состоящие из одного медицинского прибора. В общем случае при функционировании СМО каждая из подсистем может находиться в одном из следующих состояний:

1) c_0 – прибор u свободен;

2) c_1 – прибор u занят обслуживанием пациента, очереди нет;

3) c_2 – прибор u занят обслуживанием пациента, 1 заявка в очереди;

4) c_3 – прибор u занят обслуживанием пациента, 2, 3, …k–1 заявок в очереди;

5) c_k – прибор u занят обслуживанием пациента, достигнут максимальный размер очереди k*.

Граф состояний подсистемы СМО, состоящей из одного обслуживающего прибора, показан на рис. 2.

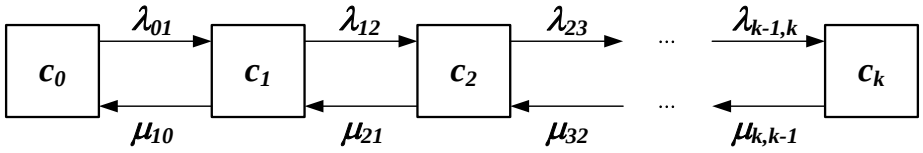


Рис. 2. Граф переходов состояний прибора СМО

Система алгебраических уравнений для определения предельных вероятностей состояний подсистемы СМО $c_0 \dots c_4$ имеет вид:

$$\begin{cases} p_0 + p_1 + p_2 + \cdots + p_k = 1 \\ p_0\lambda_{01} = p_1\mu_{10} \\ p_1(\mu_{10} + \lambda_{12}) = p_0\lambda_{01} + p_2\mu_{21} \\ p_2(\mu_{21} + \lambda_{23}) = p_1\lambda_{12} + p_3\mu_{32} \\ \cdots \\ p_k\mu_{k,k-1} = p_{k-1}\lambda_{k-1,k} \end{cases} \quad (1)$$

Поскольку число медицинских приборов, которые используются для диагностики заболеваний по нозологической форме, может быть достаточно большим, общее число состояний системы также может быть велико.

Вычисление выходного потока μ_m осуществляется исходя из знания потоков Λ_m, λ_m^s, λ_u^* и времени *t*, затрачиваемого на обслуживание (диагностику) одного пациента.

Время $\bar{t}_i$ для большинства случаев является детерминированным. Следовательно, и выходной поток μ_i также будет являться детерминированным. Поэтому *i*-ая подсистема СМО (см. рис. 1) может быть описана моделью: *M*/*D*/*1*/*k*, а вся СМО – *M*/*D*/*u*/*k*/*s*, т.е. СМО представляет собой *u* независимо функционирующих подсистем [3].

Характеристики *i*-й подсистемы СМО определяются по формулам:

1) нагрузка *i*-й подсистемы СМО – $\rho_i^* = \lambda_i^* \bar{t}_i$;

2) производительность *i*-й подсистемы СМО – $A = \lambda_i^* p_0$, где p_0 – вероятность того, что заявка будет обслужена, т.е. не потеряна, считается достаточно высокой;

3) среднее время ожидания заявки в очереди в *i*-й подсистеме СМО – $\bar{t}_{\text{оч}} = \frac{\rho_i^*}{2(1-\rho_i^*)}$.

Суммарная нагрузка СМО – $P = \sum_{i=1}^{u} \rho_i^*$.

Среднее время ожидания заявок в СМО – $\bar{T}_s = \sum_{s=1}^{4} \xi_s \bar{t}_s$, где $\xi_s = \frac{\lambda_m^s}{\Lambda_m}$ – коэффициент, учитывающих долю заявок класс *s* в суммарном потоке, который может трактоваться, как вероятность того, что поступившая в систему заявка принадлежит классу *s*.

После окончания диагностики пациента он покидает соответствующий медицинский прибор и систему с вероятностью *W*, или с вероятностью *R* направляется на дополнительное исследование с помощью другого прибора. Вероятности *W* и *R* определяются показателем: «Усредненный показатель частоты предоставления», – устанавливаемым стандартом оказания медицинской помощи по соответствующей нозологии.

Распределение вероятностей нахождения системы в состоянии C_i описывается матрицей переходных вероятностей:

$$C = \begin{vmatrix} p_{10}\, p_{11}\, p_{12} \cdots p_{1k} \\ p_{20}\, p_{21}\, p_{22} \cdots p_{2k} \\ \cdots \\ p_{u0}\, p_{u1}\, p_{u2} \cdots p_{uk} \end{vmatrix}, \quad (3)$$

причем $\sum_u p_{uk} = 1$.

Время нахождения пациента с определенной степенью тяжести заболевания определяется выражением:

$$\bar{T}_{\text{сист}} = \sum_{s=1}^{4}[\xi_s(t_i + t_{\text{оч}_i})]. \quad (4)$$

Определение параметра системы μ_m осуществляется исходя из $\bar{t}_i$:

$$\mu_m = \sum_{\text{i}=1}^{\text{u}} \mu_{\text{i}} = {}^{1}/_{\sum_{i=1}^{u} \bar{t}_i} \quad (5)$$

Таким образом, система массового обслуживания, являясь моделью процесса диагностирования заболевания, позволяет получить оценки основных параметров подсистемы диагностики лечебно-диагностического процесса. Знание этих оценок, в свою очередь, является важной составляющей процесса имитационного моделирования функционирования медицинской организации при оказании медицинской помощи.

Библиографический список

1. Шувалова, Е. П. Ошибки в диагностике инфекционных болезней / Е. П. Шувалова. – URL: http://www.medchitalka.ru (дата обращения: 20.02.2015).

2. Кобринский, Б. А. Логика аргументации в принятии решений в медицине / Б. А. Кобринский // Журнал НТИ, Сер. 2. – 2001. – № 9. – С. 1–8.

3. Алиев, Т. И. Основы моделирования дискретных систем / Т. И. Алиев – СПб. : СПбГУ ИТМО, 2009. – 363 с.

Е. В. Чернова, П. Н. Полежаев

Оренбургский государственный университет, Оренбург, Россия

АНАЛИЗ СУЩЕСТВУЮЩИХ ТЕХНОЛОГИЙ НЕЙРОННЫХ СЕТЕЙ

Рассматриваются современные библиотеки и фреймворки для работы с нейронными сетями, а также существующие облачные платформы, позволяющие проводить глубокое машинное обучение. Их возможности анализируются и сравниваются для выявления наиболее приемлемого варианта.

В последнее время идея использования нейронных сетей и машинного обучения в различных сферах жизни человека приобрела необычайную популярность. Активно разрабатываются системы распознавания различных объектов по фотографиям, прогнозирования событий и даже сочинения стихотворений. Уже существуют инструменты, такие как NeuroSolutions for MATLAB и STATISTICA Neural Networks, реализующие функции анализа, управления, визуализации данных, а также сервисы развлекательного характера, использующие нейронные сети.

Цель данной статьи определить наиболее подходящие технологии для решения фундаментальной научной проблемы повышения производительности распределенных нейронных сетей в облачных системах с гибридной архитектурой.

В настоящее время имеется ряд разработок, представляющих собой высокоуровневые инструментальные комплексы проектирования прикладных систем, использующих нейронные сети. Математические пакеты Wolfram и Matlab обрабатывают данные с учетом моделирования и прогнозирования, они могут работать с нейронными сетями. Например, Neural Network Package от Wolfram поддерживает сети прямого распространения (Feed Forward Neural Networks, FFNN), сети радиально-базисных функций (Radial Basis Function network, RBFN) и сети Хопфилда (Hopfield Network, NH). Пакет расширения MATLAB Neural Network Toolbox также работает с рекуррентными нейронными сетями (Recurrent

Neural Network, RNN) [1], кроме того он поддерживает параллельные вычисления с помощью Parallel Computing Toolbox.

Подробнее рассмотрим и сравним некоторые библиотеки и фреймворки, позволяющие работать с нейронными сетями и отвечающие следующим условиям:

- открытая лицензия программного обеспечения – это позволит в будущем изменять или расширять ПО;
- хорошо продуманная и доступная документация.

Важным критерием сравнения является поддержка нескольких типов нейронных сетей. Особый интерес представляют сверточная (англ. Convolutional Neural Network, CNN), рекуррентная, рекуррентная сверточная(англ. Recurrent Convolutional Neural Network, RCNN) нейронные сети, сеть прямого распространения, глубокая сеть доверия (Deep Belief Network, DBN), ограниченная машина Больцмана (Restricted Boltzmann Machine, RBM), а также разновидность рекуррентной нейронной сети – долгая краткосрочная память (Long Short-Term Memory, LSTM).

Основным ограничением развития нейронных сетей часто признают высокие вычислительные затраты на их реализацию. Один из методов решения данной проблемы – использование параллельных и распределенных вычислений на специализированном аппаратном обеспечении. Наиболее ярким примером является программно-аппаратная вычислительная архитектура параллельных вычислений CUDA компании NVIDIA. CUDA помогает реализовывать алгоритмы, исполняемые на графических процессорах видеоускорителей. OpenCL – еще один стандарт для разработки приложений для гетерогенных систем, он также используется для параллельных вычислений на центральных и графических процессорах, в том числе и под управлением CUDA [2].

Наличие предварительно обученных моделей является большим плюсом библиотеки, поскольку их можно применять в своих приложениях, экономя время и ресурсы. Чтобы лучше адаптировать модель к новым данным, можно провести дополнительную настройку – обучить нейронную сеть с помощью собственного набора данных.

В сравнительной табл. 1 также указаны платформы, на которых библиотеки способны работать, и их программные интерфейсы (API).

Таблица 1

Сравнение библиотек для нейронных сетей

Название ПО	Платформа	API	Типы нейросетей	Поддержка GPU	Параллельность	Обученные модели
1	2	3	4	5	6	7
TensorFlow	Linux, Mac OS X, Windows	C++, Python	CNN, RNN, DBN, RBM	CUDA	CPU, GPU	TensorFlow Models
Theano	Linux (Windows, Mac)	Python	CNN, RNN, DBN, RBM	CUDA, OpenCL	GPU	С помощью Model Zoo в Lasagne
Lasagne	Linux, Mac (Windows)	Python	CNN, RNN, LSTM	CUDA, cuDNN	GPU	Использует модели Caffe
CNTK	Windows, Linux	Python, C++, C#, .NET, Java	CNN, RNN, LSTM, FFNN	CUDA, OpenMP	CPU, GPU	Model Gallery
Keras	Linux, Mac OS X, Windows	Python	CNN, RNN, DBN, RBM	CUDA	GPU (с Tensor-Flow)	Keras Application
MXNET	Linux, Mac for R and Python, Windows only for R	R, Python, Julia, MATLAB, Scala и Javascript	CNN, RCNN, LSTM, DBN, RBM	CUDA	CPU, GPU	Инструмент, преобразующий модели Caffe в формат MXNET

Окончание табл. 1

1	2	3	4	5	6	7
Deeplearning4j	Linux, Mac OS X, Windows, Android	Java, Scala, Clojure, Python (Keras)	CNN, RNN LSTM, DBN, RBM	CUDA, OpenMP	GPU	Импорт моделей из Theano, Caffe, Torch, TensorFlow, используя Keras
Caffe	Linux, Mac OS X, Windows	Python, MATLAB	CNN, RCNN, LSTM	CUDA, OpenMP	GPU	Model Zoo
Torch	Linux, Mac OS X, iOS Windows, Android,	Lua, C, C++	CNN, RNN, DBN, RBM	CUDA, OpenCL	CPU, GPU	Импорт моделей с помощью LoadCaffe

Самыми быстрыми среди рассмотренных библиотек являются Theano и Torch, использующий малоизвестный язык Lua. Однако, Theano – низкоуровневый фреймворк и больше рассматривается как исследовательская платформа, а не библиотека глубокого обучения [3]. Theano часто используется вместе с библиотеками, имеющими более высокий уровень абстракции, такими как Keras и Lasagne.

Caffe хорошо известен из-за своего набора предобученных моделей нейронных сетей Model Zoo, который могут использовать другие фреймворки и библиотеки с помощью дополнительных инструментов [4]. Но сам фреймворк слишком громоздкий для больших сетей, он не так хорош для работы с рекуррентными сетями и медленно развивается.

Microsoft Cognitive Toolkit, также известный как CNTK, не годится для коммерческого использования, так как имеет разрешительную лицензию, однако он быстрее чем TensorFlow [5].

MXNET – отличное решение для любителей языка R, единственная платформа, которая поддерживает все его функции и, кроме того, обладает высокой производительностью и эффективным использованием памяти [6].

Deeplearning4j создан для использование в бизнес-среде, адаптирован для микросервисной архитектуры, расширяется за счет Hadoop [7]. Может использовать обратное распространение по времени (Backpropagation Through Time, BPTT) – основанную на градиентах технику тренировки определенных типов рекуррентных нейронных сетей.

Тем не менее самой популярной считается библиотека TensorFlow, хотя она медленнее, чем Torch и Theano и не имеет такого большого набора моделей, как Caffe. В отличие от любой другой архитектуры, TensorFlow имеет возможность делать частичные вычисления подграфа, то есть получение выборки от общей нейронной сети, а затем ее обучение отдельно от остальной части. Это так называемое Model Parallelization, которое используется для распределенного обучения. Caffe также поддерживает Keras и обладает обширной документацией, которая охватывает теоретические основы и практическое применение нейронных сетей.

Таким образом, несмотря на преимущества прочих библиотек, для выполняемой НИР была выбрана библиотека TensorFlow, как надежный и развивающийся проект, способный работать с разными типами нейронных сетей и поддерживающий параллельные вычисления с помощью графических процессоров, а также Keras в качестве высокоуровневого API для TensorFlow.

Использование графических ускорителей даже с самыми последними драйверами не всегда могут обеспечить быстрое обучение и эксплуатацию больших нейронных сетей. Для них требуются значительные вычислительные ресурсы и, кроме того, огромные дисковые пространства, поскольку при обучении объем нейронной сети увеличивается. Однако существует другой подход – использование облачных вычислений. Основным их преимуществом является обеспечение пользователя сервисами вычисления с высокой надежно-

стью, удобством масштабирования по запросу и гибкостью в развертывании и настройке решения [8]. Такие гиганты как Google, Amazon и Microsoft предоставляют подобные услуги.

Amazon Web Services CloudFormation упрощает процесс создания распределенных кластеров глубокого обучения. CloudFormation использует Amazon Deep Learning AMI для настройки кластера экземпляров Amazon EC2 и других AWS ресурсов необходимых для осуществления распределенного глубокого обучения. В свою очередь Amazon Deep Learning AMI обеспечивает надежную, защищенную и высокопроизводительную среду для глубокого обучения приложений. Он включает в себя такие фреймворки, как MXNet, Caffe, TensorFlow, Theano, CNTK, Torch и Keras, а также многие популярные библиотеки AWS. Deep Learning AMI предоставляется без дополнительной платы пользователям Amazon EC [9].

Google Cloud Machine Learning обеспечивает современные сервисы машинного обучения с предобученными моделями и сервисом для генерации своих собственных адаптированных моделей [10]. Обладает огромными возможностями, в том числе создание огромной портативной масштабируемой модели, способной интегрироваться с другими продуктами Google (Cloud Storage, Cloud Dataflow, Cloud Datalab). Предлагаются готовые решения для анализа видео, изображений, текста и разговорной речи. Для работы рекомендуется использовать TensorFlow.

Microsoft Azure предоставляет набор приложений для машинного обучения Machine Learning. Простой интерфейс и наличие образцов в Cortana Intelligence Gallery помогают создавать собственные решения [11]. Преимуществами являются неограниченное дисковое пространство и количество модулей на каждый эксперимент, высокая производительность, возможность задействовать несколько узлов, наличие соглашения об уровне обслуживания.

Наиболее гибкой платформой в ценовом плане является AWS CloudFormation, она же обладает огромным количеством дополнительных инструментов и готовых моделей нейронных сетей, поэтому для дальнейшего исследования предпочтительно ее использование.

Исследование проведено при поддержке Правительства Оренбургской области и РФФИ (грант №16-29-09639 и № 17-47-560046), Президента Российской Федерации, стипендии для молодых ученых и аспирантов (СП-2179.2015.5).

Библиографический список

1. Медведев, В. С. Нейронные сети MATLAB 6 / В. С. Медведев, В. Г. Потемкин. – М. : Диалог-МИФИ, 2002.

2. Парубец, В. В. Применение технологии CUDA для ускорения вычислений в нейронных сетях / В. В. Парубец, О. Г. Берестнева, Д. В. Девятых // Известия Томского политехнического университета. – 2012. – Т. 320, № 5.

3. Bastien, F. Theano: new features and speed improvements / F. Bastien et al. // arXiv preprint arXiv:1211.5590. – 2012.

4. Jia, Y. Caffe: Convolutional architecture for fast feature embedding / Y. Jia et al. // Proceedings of the 22nd ACM international conference on Multimedia. – ACM, 2014. – С. 675–678.

5. Reasons to Switch from TensorFlow to CNTK. – URL: https://docs.microsoft.com/en-us/cognitive-toolkit/reasons-to-switch-from-tensorflow-to-cntk (дата обращения: 23.06.2017).

6. Chen, T. Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems / T. Chen et al. // arXiv preprint arXiv:1512.01274. – 2015.

7. Deep Learning for Java. – URL: https://deeplearning4j.org (дата обращения: 22.06.2017).

8. Xu, X. From cloud computing to cloud manufacturing / X. Xu // Robotics and computer-integrated manufacturing. – 2012. – Т. 28, №. 1. – С. 75–86.

9. Distributed Deep Learning Made Easy. – URL: https://aws.amazon.com/ru/blogs/compute/distributed-deep-learning-made-easy/ (дата обращения: 24.06.2017).

10. Cloud Machine Learning Services. – URL: https://cloud.google.com/products/machine-learning/ (дата обращения: 24.06.2017).

11. Chappell, D. Introducing Azure Machine Learning / D. Chappell // A guide for technical professionals, Sponsored by Microsoft Corporation. – 2015.

А. И. Мартышкин

Пензенский государственный технологический университет, Пенза, Россия

РЕАЛИЗАЦИЯ ОПЫТНОГО ОБРАЗЦА РЕКОНФИГУРИРУЕМОЙ ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЫ ДЛЯ ЦИФРОВОЙ ОБРАБОТКИ СИГНАЛА НА БАЗЕ ПРОГРАММИРУЕМЫХ ЛОГИЧЕСКИХ ИНТЕГРАЛЬНЫХ СХЕМ

Целью работы является проведение обзора современных российских реконфигурируемых вычислительных систем. Приводятся описание и результаты работы опытного образца реконфигурируемой вычислительной системы для цифровой обработки сигнала на базе ПЛИС. В заключении приведены полученные результаты и выводы.

Разработка высокопроизводительных вычислительных систем (ВВС), является наиважнейшей задачей в сфере современной науки и техники. Главный вектор развития данного направления – наращивание производительности вычислительных комплексов. Реконфигурируемая вычислительная система (РВС) представляет собой перестраиваемую архитектуру под определенную задачу, в которой объединены реконфигурируемые вычислительные блоки (РВБ), на базе программируемых логических интегральных схем (ПЛИС), с управляющими процессорами. В России разрабатываются и используются следующие РВС: прежде всего устройства РВС-1, РВС-0.2-РС и РУПК-50, созданные в НИИ МВС ЮФУ; вычислитель БВР-01, разработанный в компании Fastwel; устройство КВ8-К7-410-01 «Топаз-2», разработанный ФГУП НИИ «Квант», реконфигурируемая вычислительная плата RC-47, спроектированная в ООО НПО «Роста» и др. [1].

В НИИ МВС ЮФУ ведутся разработки по направлению многопроцессорных систем с перепрограммируемой архитектурой, в рамках этих разработок созданы РВС, состоящие из РВБ РВС-0.2-ВБ (рис. 1,*а*), построенных на основе базового модуля 16*V*5-75 «Алькор» (рис. 1,*б*), он так же предназначен для комплектования рабочей станции РВС-0.2-РС (рис. 1,*в*). С помощью рабочей станции и вычислительного блока организуются старшие представители семейства РВС-1Р и РВС-5. 16 базовых блоков, укомплектованных в РВС-5, дают производительность в порядке 1000 Гфлопс [1]. Базовый модуль 16*V*5-75 «Алькор» представляет собой печатную плату, размещенную на ней 16 ПЛИС серии *Virtex*5 фирмы *Xilinx*. Архитектура модуля 16*V*5-75 может реализовать 256 устройств с плавающей запятой с 32-разрядными форматом данных.

Еще одна разработка НИИ МВС ЮФУ – ускоритель ЭВМ РУПК-25 (рис. 1,*г*) для решения трудоемких задач с производительностью 25 Гфлопс. Для комплектования РУПК-25, используется базовый модуль 16S3-25 (рис. 1,*д*), представляющий собой плату с размещенными на ней 16 ПЛИС серии *Spartan* 3 фирмы *Xilinx*. Модуль имеет возможность реализовать 128 устройств с плавающей запятой с 64-разрядным форматом данных или 256 устройств с плавающей запятой с 32-разрядным форматом данных, связанных между собой пространственной коммутационной системой [1].

Современные тенденции развития микроэлектронных технологий позволяют применять архитектурные решения, значительно повышающие производительность ВВС. РВС, описанные выше, являются перспективными разработками в области высокопроизводительных вычислений [2, 3].

Данное технологическое решение заключается, в применении ПЛИС вместо процессорных узлов, что дает возможность построения гибкой системы. Подобное решение используется при реализации аппаратного диспетчера задач, входящего в состав описываемой реконфигурируемой вычислительной системы. В нашем случае РВС – это устройство, состоящее из 4 процессоров, реализованных на ПЛИС, для цифровой обработки сигнала.

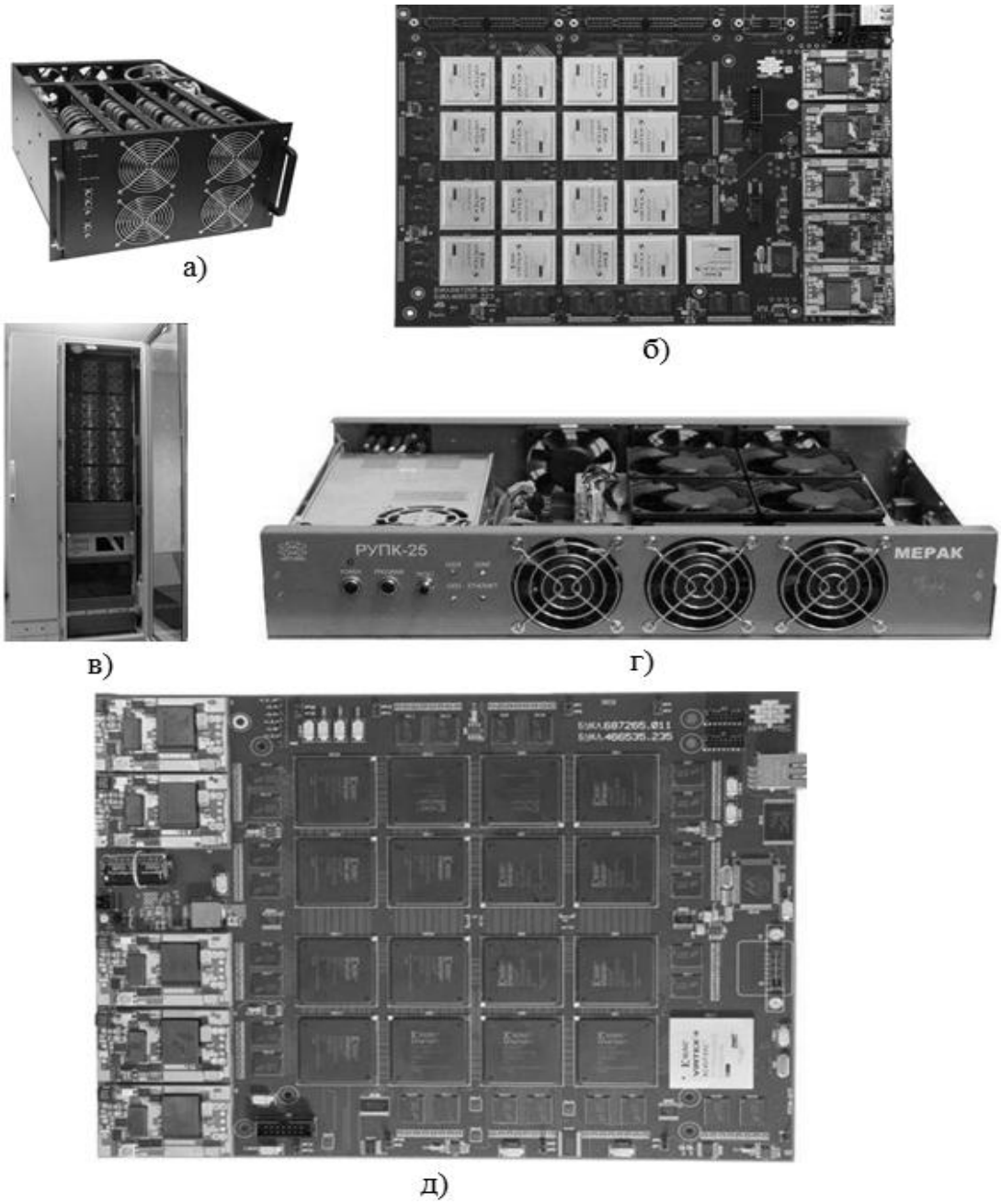


Рис. 1. Вычислительный блок РВС-0.2-ВБ (*а*); базовый модуль 16V5-75 «Алькор» (*б*); система РВС-1Р (*в*); ускоритель персонального компьютера РУПК-25 (*г*); базовый модуль 16S3-25 «Мерак» (*д*)

Разработка подобной системы на ПЛИС предоставляет возможность для перенастройки устройства под различные классы задач ЦОС. На рис. 2,*а* представлена структурная схема проектируемой РВС.

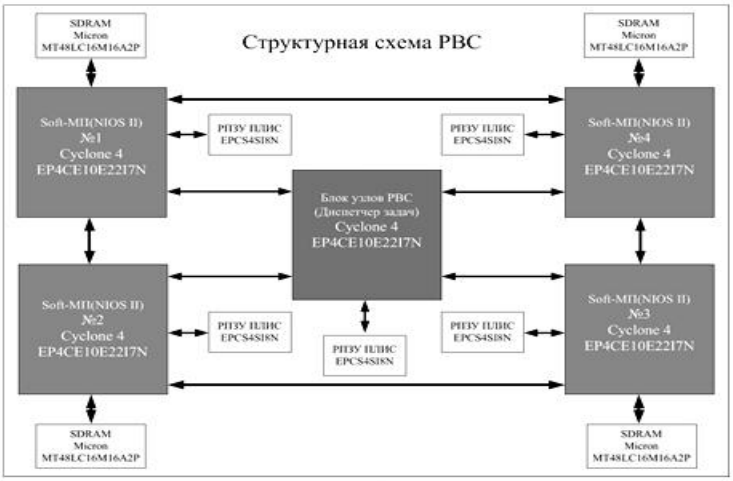
Разрабатываемая РВС состоит из следующих компонентов [2, 3]:

– 5 ПЛИС *Cyclone* 4 *EP*4*CE*10*E*22I7*N* фирмы *Intel*, 4 ПЛИС используются для *soft*-микропроцессорных ядер *Nios* II, 1 ПЛИС выделена под блок узлов системы;

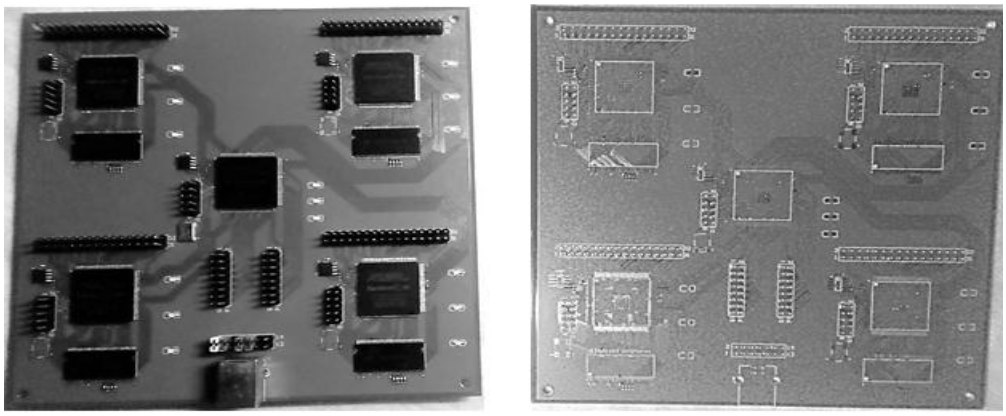
– 5 РПЗУ ПЛИС *EPCS*4*SI*8*N* фирмы *Intel*, для хранения конфигурации каждой ПЛИС;

– 4 *SDRAM MT*48*LC*16*M*16*A*2 фирмы *Micron*, блоки ОЗУ.

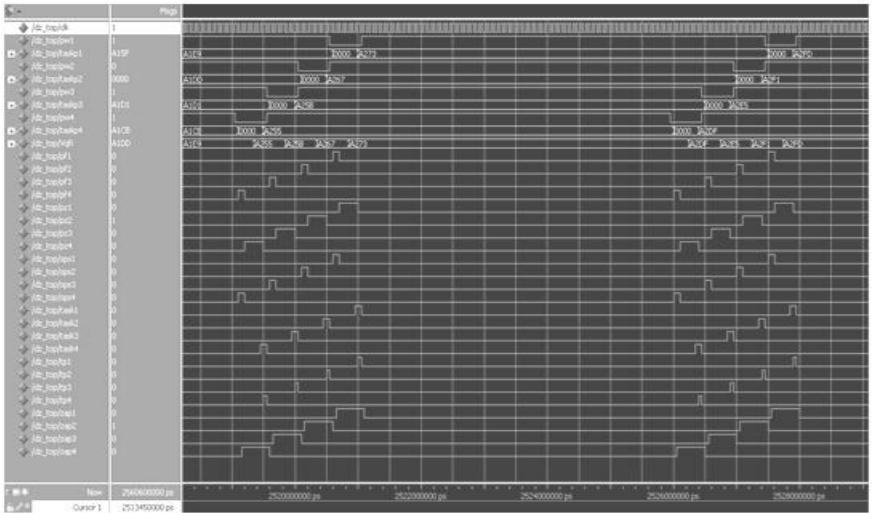
На основе представленной на рис. 2,*а* схемы, разработаны принципиальная схема и схема печатной платы, по которым был создан опытный образец устройства, представленный на рис. 2,*б*.



а)



б)



в)

Рис. 2. Структура разрабатываемой РВС (*а*); плата готового опытного образца РВС (*б*); временные диаграммы работы системы в среде моделирования ModelSim-Altera 10.0c (*в*)

Для проведения эксперимента с спроектированной РВС, созданы блоки имитации работы 4 процессоров и блок генератора задач. Моделирование проводилось с помощью ПО *ModelSim-Altera* 10.0*c*. После компиляции была запущена симуляция проекта, где были получены временные диаграммы (рис. 2,*в*) работы РВС, которые показывают, что идентификатор поступившей задачи под номером *A*255 передался на обработку в процессор под номером 4 (*taskp*4), после чего процессор выставил сигнал о том, что принял

задачу на обработку (*tp*4). Процессор под номером 4 оказался занятым, поэтому следующий идентификатор задачи *A*25B принят на обработку в третий процессор. Подобно четвертому процессору, третий выставил сигнал (*tp*3) о том, что приступил к обслуживанию задачи. По результатам моделирования видно, что первым для обслуживания текущей задачи назначен четвертый процессор, следом назначается третий процессор, и так далее. Это обусловлено схемой приоритетов от выполнения задач в системе, т.е. от четвертого процессора к первому.

Обзор по архитектурам отечественных РВС показал, что нынешние тенденции наращивания производительных мощностей ВВС, например, увеличение числа транзисторов на кристалле, постепенно исчерпывают себя. Подход к построению РВС позволяет применять преимущества новых технологий и распараллеливания процессов одновременно. За счет бурного роста развития микроэлектроники и переход к технологическому процессу до 9 нм РВС сегодня представляют широкий интерес разработчиков ВВС.

Аппаратная реализация диспетчера задач полностью снимает проблему временных потерь в период распределения и синхронизации процессов в РВС. Область применения полученных результатов возможно не только в данном проекте, но и в других аналогичных системах, где важно повышение производительности и эффективности.

Библиографический список

1. Левин, И. И. Высокопроизводительные реконфигурируемые вычислительные системы на основе ПЛИС VIRTEX-7 / И. И. Левин, А. И. Дордопуло, И. А. Каляев, В. А. Гудков // Параллельные вычислительные технологии (ПаВТ'2014) : сб. тр. Междунар. науч. конф. / Л. Б. Соколинский, К. С. Пан. – Челябинск : Изд. центр ЮУрГУ, 2014. – С. 131–139.

2. Мартенс-Атюшев, Д. С. Реконфигурируемый вычислительный кластер для цифровой обработки сигнала / Д. С. Мартенс-Атюшев, А. И. Мартышкин // Современные методы и средства обработки пространственно-временных сигналов : сб. ст. XIII Всерос. науч.-техн. конф. / под ред. И. И. Сальникова. – Пенза, 2015. – С. 112–117.

3. Мартенс-Атюшев, Д. С. Разработка и исследование реконфигурируемого вычислительного кластера для цифровой обработки сигнала / Д. С. Мартенс-Атюшев, А. И. Мартышкин // Современные информационные технологии. – 2015. – № 21. – С. 190–195.

В. А. Романчук

Рязанский государственный университет им. С. А. Есенина, Рязань, Россия

ПОВЫШЕНИЕ ПРОИЗВОДИТЕЛЬНОСТИ НЕЙРОКОМПЬЮТЕРНЫХ УСТРОЙСТВ ЗА СЧЕТ УПАКОВКИ ДАННЫХ

В настоящее время достаточно актуальна проблема недостаточности вычислительных ресурсов для ряда задач, таких как эмуляция нейронных сетей, распознавание образов и т.д. Сегодня эту проблему решают путем применения систем и технологий распределенных вычислений, в том числе с использованием нестандартных, не «фон-Неймановских» вычислительных архитектур. В работе показан способ ускорения выполнения программ с высокой степенью нейробазисных вычислений за счет параллельного выполнения операций с малоразрядными операндами.

В работе предлагается способ увеличения производительности нейрокомпьютерных программ за счет так называемой упаковки данных и результатов операций. Эффектив-

ность способа достигается за счет параллельного выполнения операций с малоразрядными операндами. В некоторых моделях нейропроцессоров существует возможность аппаратной упаковки данных.

Каждый класс задач $Z^{(j)}$, представленный в виде ИНС обладает набором характеристик:

Пусть $NetX_{R_i}$ – разрядность i-го входа ИНС $Net_X = \{net_{x1},....net_{xi},...,net_{xn}\}$, определяемая по его максимально возможному значению:

$$NetX_{R_i} = \log_2(\max(net_{xi}));\quad (1)$$

$NetW_{R_i}$ – разрядность i-го весового коэффициента ИНС $W = \{w_{w1},....w_{wi},...,w_{nw}\}$, определяемая по его максимально возможному значению;

$$NetW_{R_i} = \log_2(\max(w_{xi}));\quad (2)$$

где $NetY_{R_i}$ – разрядность i-го выхода ИНС $Net_Y = \{net_{y1},....net_{yi},...,net_{yn}\}$, определяемая по его максимально возможному значению:

$$NetY_{R_i} = \log_2(\max(net_{yi}));\quad (3)$$

где где $H_{\Pr}$ – разрядность нейропроцессора.

Тогда, при выполнении эмуляции нейрона, в нейропроцессоре должно быть зарезервировано определенное количество бит, равное разрядности максимально возможной разрядности результата, которая определяется как сумма разрядностей входных данных и одного бита для переноса:

$$R = \max(NetX_{R_i}) + \max(NetW_{R_i}) + 1.\quad (4)$$

В этом случае, количество параллельных потоков данных Pt за счет упаковки данных определяется как:

$$Pt = \frac{H_{\Pr}}{\max(NetX_{R_i}) + \max(NetW_{R_i}) + 1}.\quad (5)$$

Для вещественных значений входных значений, весовых коэффициентов рассматривается отдельно сумма разрядностей целых и дробных частей.

Описанный способ может быть использован для упрощения упаковки данных, но не обеспечивает возможность битового перераспределения операции, в связи с чем предлагается второй, более трудозатратный для программиста способ. В этом случае, цель организации обработки информации заключается в перераспределении множества операций и, соответственно, множества микрокоманд так, чтобы максимальное количество операций было выполнено за один такт за счет упаковки данных.

Для каждого кортежа операций $SO_i =< O_{i1}, O_{i2},..., O_{inc} >$ можно определить кортеж разрядностей вычислений для каждой j-й операции $< R_{i1}, R_{i2},..., R_{iinc} >$ следующим образом:

$$R_i = NetX_{R_i} + NetW_{R_i} + 1.\quad (6)$$

То есть необходимо перераспределить множество операций в кортежи операций так, что разность между разрядностью нейропроцессора и суммарной разрядностью вычислений была минимальна:

$$SO_i =< O_{i1}, O_{i2},..., O_{inc} >= SO_{i1} \cup SO_{i2} \cup ... \cup SO_{iT},$$

$$SO_{ij} =< O_{ij1}, O_{ij2},..., O_{ijinc} >, j = \overline{1,T},\quad (7)$$

$$(H_{\Pr} - \sum_{k=1}^{ijinc} R_{ijk}) \to \min .$$

где T – количество тактов нейропроцессора для реализации ИНС.

Каждая операция O_{ijk} обладает целочисленной характеристикой разрядности вычисления, равной R_{ijk}.

Тогда, кортеж операций SO_{ij} может быть выполнен за один такт нейропроцессора, обеспечивая эмуляцию $ijinc$ нейронов.

Задача упаковки данных в нейропрограмме может быть описана математически следующим образом:

$$\forall SO_i, i = \overline{1, p}; \sum_{j=1}^{T} (\Pr - \sum_{k=1}^{ijinc} R_{ijk}) \to \min . \quad (8)$$

Задача (8) NP-полная задача оптимизации, по существу, сводимая к задаче целочисленного линейного программирования – задаче раскроя. В начале становления научных исследований в области снижения «расходов» в целевых функциях большинство из оптимизационных методов использовали линейные модели и методы линейного программирования. Основы данных методов были заложены работами Л. В. Канторовича, В. А. Залгаллера и P. Gilmore, R. Gomory [1]. Также получили развитие точные методы, использующие методологию «ветвей и границ», описанные в работах И. В. Романовского, С. В. Канцева и S. Martello, D. Toth, а также их дальнейшие исследования, которые велись в области уменьшения объема перебора [2]. Для решения этих задач часто используются простые однопроходные и многопроходные эвристики, а также метаэвристические алгоритмы, например генетический алгоритм или алгоритм поиска с запретами. В данной работе предлагается комбинированный оптимизационный подход к решению задачи одномерной оптимизации с учетом характеристики разрядности нейропроцессора, основанный на методах динамического программирования.

Введем понятие остатка разрядности, определяющего разность между характеристикой разрядности нейропроцессора и суммарной разрядностью вычислений для всех операций, уже входящих в кортеж $Ost_j = \Pr - \sum_{k=1}^{ijinc} R_{ijk}$.

На первом шаге производится первичное заполнение всех кортежей по следующему правилу: в кортеж, имеющий наибольший остаток разрядности, помещается операция с наибольшей разрядностью вычисления.

На втором шаге выполняется попарный обмен операциями в кортежах, если это приводит к уменьшению остатка в наиболее заполненном кортеже (рис. 1). При каждом обмене операциями выполняется сортировка всех кортежей по остатку, а также операций в кортеже от большей к меньшей.

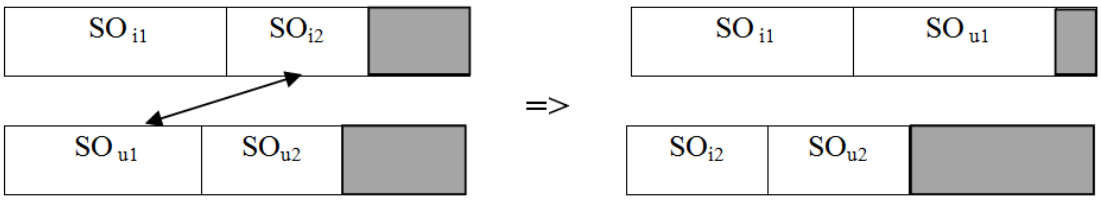


Рис. 1. Попарный обмен операциями между кортежами

Для уменьшения остатков раскроя на шаге 2 помимо попарного обмена деталями выполняется обмен тремя операциями в кортежах, если это приводит к уменьшению остатка в наиболее заполненном кортеже (рис. 2). Данное дополнение позволяет

значительно уменьшить суммарный остаток, но увеличивает время выполнения алгоритма в зависимости от количества операций в условиях задачи.

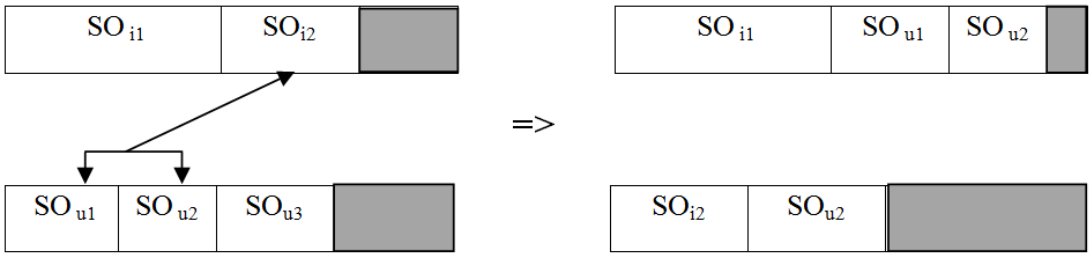


Рис. 2. Тройной обмен операциями между кортежами

Для оценки эффективности распараллеливания на данном этапе введем коэффициент, показывающий долю загруженности ядра нейропроцессора при выполнении u-й операции:

$$K_{Rju} = 1 - \frac{\mathrm{Pr} - \sum_{k=1}^{ujinc} R_{ujk}}{\mathrm{Pr}}, \tag{9}$$

Суммарные показатели загруженности ядра нейропроцессора для фрагмента SO_j:

$$K_{Rj} = 1 - \frac{\sum_{j=1}^{T}(\mathrm{Pr} - \sum_{k=1}^{ijinc} R_{ijk})}{T * \mathrm{Pr}}. \tag{10}$$

Для микронейропрограммы в целом:

$$K_R = 1 - \frac{\sum_{i=1}^{p}\sum_{j=1}^{T}(\mathrm{Pr} - \sum_{k=1}^{iujinc} R_{iujk})}{p * T * \mathrm{Pr}}. \tag{11}$$

Следует отметить, что описанный математический аппарат эффективен только при малоразрядных входных, выходных данных и весовых коэффициентах. При малоразрядных только входных данных или весовых коэффицинтах возникнут простои, что снизит эффективность организации обработки информации.

Библиографический список

1. Канторович, Л. В. Расчет рационального раскроя материалов / Л. В. Канторович, В. А. Залгаллер. – Ленинград : Лениздат, 1951. – 198 с.

2. Крючковский, В. В. Детерминизация модели многофакторного оценивания при различных видах неопределенности задания параметров / В. В. Крючковский, А. В. Усов // Научный и производственно-практический сборник по техническим и естественным наукам : тр. Одесского политехн. ун-та. – Одесса, 2009. – Вып. 2 (32). – С. 154–160.

3. Романчук, В. А. Модель описания функционирования нейропроцессора как сложной параллельной системы / В. А. Романчук, В. В. Лукашенко // Автоматизация и управление в технических системах. – 2015. – № 2. – URL: auts.esrae.ru/14-266

4. Романчук, В. А. Разработка математической модели реструктуризуемого под классы задач, виртуализируемого кластера GRID-системы на базе нейропроцессоров / В. А. Романчук, В. В. Лукашенко // Вестник Рязанского государственного университета им. С. А. Есенина. – 2014. – № 1 (42). – С. 177–182.

5. Ручкин, В. Н. Обобщенная модель вычислений кластера нейрокомпьютеров / В. Н. Ручкин, В. А. Романчук, В. В. Лукашенко // Вестник Рязанского государственного университета им. С. А. Есенина. – 2015. – № 2 (47). – С. 146–150.

А. В. Осипов

Пензенский государственный университет, Пенза, Россия

АЛГОРИТМЫ, ПРИМЕНЯЕМЫЕ В ИГРАХ

Произведено исследование самых популярных игр для платформы Android OS с целью выявления наиболее распространенных алгоритмов, используемых для реализации логики игр разных жанров.

Введение.

Алгоритм – набор инструкций, описывающих порядок действий исполнителя для достижения некоторого результата. В старой трактовке вместо слова «порядок» использовалось слово «последовательность», но по мере развития параллельности в работе компьютеров слово «последовательность» стали заменять более общим словом «порядок». Независимые инструкции могут выполняться в произвольном порядке, параллельно, если это позволяют используемые исполнители [1].

Актуальность.

В настоящее время возникает большая потребность в выявление наиболее подходящего алгоритма реализации разного вида программного обеспечения для привлечения и удержания пользователей в его использования. И данная статья нацелена на поиск наиболее широко используемого вида алгоритма, где в качестве вида программного обеспечения была выбрана видеоигра для портативных устройств.

Цели, задачи.

Основной целью данной статьи стало изучение самых распространенных видов алгоритмов в играх на платформе Android OS. Исследовать какие виды алгоритмы наиболее часто используются в тех или иных игровых жанрах и выявить их достоинства и недостатки. А также рассмотреть, как данные алгоритмы влияют на деятельность человека и развитию каких навыков они способствуют.

Изложение основного материала.

Для того чтобы дать ответ на задачи, поставленные в данной статье, было следованно двадцать игр для мобильных устройств на платформе Android OS, расположенных в онлайн сервисе «Google Play Маркет». Были выбраны следующие игровые жанры: аркада, головоломка и квест, по десять игр на каждый жанр, где на текущей момент времени наиболее популярные игры являются: «2048», «100 дверей», «Timber Man», «Floppy Bird» и «Побег из комнаты».

После ознакомления с ними, была составлена табл. 1, в которой приведено то, как соотносятся виды алгоритмы с игровыми жанрами.

Таблица 1

Сопоставление игровых жанров

Жанр / Алгоритм	Аркада	Головоломка	Квест
Линейный	3	8	0
Разветвляющийся	1	2	10
Циклический	6	0	0

Из данной таблице видно, что для игрового жанра аркада распространенным видом алгоритма является – циклический, а разветвляющийся, напротив, самый малоиспользуемый. Это объясняется тем, что игры такого жанра, рассчитанные на то, что пользователь оттачивает какую-либо последовательность своих действий и многочисленно это повторя-

ет, стремясь улучшить свой предыдущий результат. Для жанра головоломка ключевым видом алгоритмом, для реализации концепции игры, является линейный, так как в играх такого жанра, как правило всегда имеется одна строго последовательность действий, по завершению которой должен быть один конкретный правильный результат. Жанр квест, по результатам анализа, использует только разветвляющий алгоритм. Это можно объяснить тем, что при прохождении игр данного жанра, используется игровая механика, ориентируемая на то, что пользователю всегда предоставляется выбор из возможных вариантов решения той или иной задачи, по результатам которых строится цепочка решений, которая введет к одной из многих концовок игры. По данным онлайн сервиса «Google Play Маркет», самое большое количество загрузок приходятся на игры жанра «Аркада», где ключевой алгоритм является линейный. Из этого можно сделать вывод, что именно линейный алгоритм построения концепции мобильных игр, является самым востребованным.

Для дальнейшего изучение алгоритмов так же необходимо исследовать их основные достоинства и недостатки (табл. 2).

Таблица 2

Достоинства и недостатки алгоритмов		
	Достоинства	Недостатки
Линейный алгоритм	Простота реализации и отладки ПО, не требует больших объемов памяти	Примитивное ПО
Разветвляющийся алгоритм	Более богатый функционал ПО, вариативность действий	Требует больших объемов памяти, возрастает сложность в реализации и отладки ПО
Циклический алгоритм	Не требует больших объемов памяти, простота реализации и отладки ПО	Бедный функционал ПО, повторяющаяся последовательность действия быстро надоедает

По данной таблице можно сделать вывод, что с точки зрения разработки программного обеспечения и его поддержания наиболее подходящий жанр является головоломка, где в большинстве случае используется линейный алгоритм, не требующий больших затрат при создании. А с точки зрения игрового процесса и долгосрочного привлечения пользователя в игру, наиболее подходящим жанром можно назвать квест, использующий разветвляющий алгоритм. Игровой жанр аркада, использующий циклический алгоритм, расположен между ними и является наиболее популярным среди всех, но он обладает существенным недостатком, из-за своей простоты в механике игры он быстро теряет интерес пользователей, из-за чего возникает необходимость в создании постоянно новых небольших проектов.

Каждый из трех видов видов алгоритмов находит свое отражение и в повседневной длительности людей, тем самым игры данного рода так же влияют и на формирование определенных навыков у пользователей. Циклический алгоритм встречается в любой рутиной работе, где человек постоянно повторяет какую-либо последовательность действий, и можно предположить, что игры, в которых применяется данный алгоритм, помогают более быстрее адаптироваться к данной вида длительности и способствуют более быстрому оттачивания необходимых навыков. Линейные алгоритмы чаще можно увидеть в научных областях, где необходимо провести строго линейные действия для достижение конкретных результатов. А разветвляющийся алгоритмы применяются в программировании, где необходимо создать проект, в котором предусмотрено множество разных функций, которые в свою очередь, могут вести к разным результатам.

Выводы.

Исходя из проведенных исследований можно сделать вывод, что самый распространенный вид игрового жанра, среди мобильной платформы Android OS, является аркада. И, следовательно, самый используемый вид алгоритма – циклический. Это можно объяснить тем, что большинство населения заняты не научной деятельностью или инженерной деятельностью, а трудовой, в которой человек постоянно делает похожую последовательность действий. И игры, такого рода, по своей концепции схожи с данным видом длительности.

Библиографический список

1. Википедия – Алгоритм. – URL: https://ru.wikipedia.org/wiki/Алгоритм (дата обращения: 15.06.2017).

В. В. Лукашенко, В. А. Романчук

Рязанский государственный университет им. С. А. Есенина, Рязань, Россия

РАЗРАБОТКА АЛГОРИТМА СИНТАКСИЧЕСКОЙ ОБРАБОТКИ НЕЙРОБАЗИСНОЙ ПРОГРАММЫ

Рассматривается вариант реализации вычислительного кластера нейрокомпьютеров. Для реализации основного принципа распределенных вычислений в статье представлен алгоритм разбиения задач, поступающих на выполнение в вычислительный кластер нейрокомпьютеров, на подзадачи. Для этого предлагается представить поступившую на выполнение в кластер программу в модифицированной постфиксной Польской записи и хранить ее в стеке команд программы. В данном разделе статьи рассматривается включение в Польскую форму других, отличных от арифметических, операторов и конструкций. На следующем шаге предлагается получить абстрактное синтаксическое дерево программы, следуя алгоритму перевода модифицированной постфиксной Польской записи из стека команд в абстрактное синтаксическое дерево.

В современной научной и производственной сфере достаточно актуальна задача использования систем распределенной обработки данных из-за недостаточности локальных вычислительных ресурсов для решения ряда задач. Для решения поставленной проблемы с достаточно высокой степенью эффективности применимы масштабируемые параллельные высокоскоростные кластеры на базе нейрокомпьютеров В работе рассматривается конкретная реализация вычислительного кластера на базе нейрокомпьютеров МВ77.07 ЗАО «НТЦ Модуль», технические характеристики, преимущества и перспективность которых описана в работах [7].

При выборе нестандартной архитектуры вычислительных машин кластера необходимо решать задачу построения модели вычислений. Модель вычислений служит связующим звеном между архитектурой вычислительной системы и моделью программирования, и в распределенных системах отражает взаимодействие процессов. Под моделью вычислений будем понимать – определение множества допустимых операций, использованных для вычисления, а также относительных издержек их применения, которые характеризуют необходимые вычислительные ресурсы – время выполнения, объем памяти, а также ограничения алгоритмов или вычислительной системы. В работе [7] указан ряд алгоритмов, разработка которых необходима для реализации модели вычислений кластера нейрокомпьютеров. Одним из таких алгоритмов является алгоритм разбиения задач на подзадачи необходимый для декомпозиции поступающих на выполнение в вычислительный кластер задач, что позволит реализовать основной принцип теории распределенных вычислений. Центральной частью алгоритма разбиения задач на подзадачи является алгоритм перевода программы, поступившей на выполнение, из польской формы, как одной из форм внутреннего представления программы в теории компиляции в абстрактное синтаксическое дерево программы, как основной структуры представляющей весь процесс передачи данных от одной операции к другой внутри самой программы.

Для успешного разбиения задачи на более мелкие, атомарные подзадачи необходимо рассматривать ее с позиции алгоритма. Каждая, поступающая на выполнение в распределен-

ный вычислительный кластер задача – есть алгоритм $Al^{(j)}$, реализованный в виде программы на языке, понятном вычислительной системе и необходимый набор данных. Выполнение любой программы $PR^{(j)}$ на вычислительной машине или группе вычислительных машин осуществляется посредством четких машинных команды, адекватно отвечающих архитектуре вычислительной системы, в виде которых представляется программа. Таким образом, любой алгоритм, записанный на любом языке программирования высокого уровня, должен быть скомпилирован в набор команд, понятных машине. В связи с этим, для успешного разбиения программы на более мелкие подпрограммы и последующего их выполнения необходимо рассмотреть ее представление с точки зрения теоретических основ компиляции и построения компиляторов.

С точки зрения теоретических основ компиляции и построения компиляторов любую программу можно представить в виде $PR^{(j)} = (Gpu, V)$, где $Gpu = (W, E, start, stop)$ – граф потока управления программы, а $V = \{v_1, ..., v_k\}$ – алфавит операторов.

Граф потока управления $Gpu = (W, E, start, stop)$ – это ориентированный граф, в котором выделены две вершины start и stop, связанные между собой посредством множеств вершин W и дуг E соответственно, и удовлетворяющий следующим требованиям:

- в start не входит ни одна дуга;
- из stop не выходит ни одна дуга;
- любая вершина достижима из start;
- из любой вершины достижима вершина stop.

Вершинами такого графа являются отдельные операции над операндами выполняющиеся независимо, а дугами являются абстрактные направления данных, необходимых для выполнения операций.

Для описания программы исполняющейся на вычислительных машинах традиционной архитектуры с позиции теоретических основ компиляции и построения компиляторов этого достаточно. В свою очередь, для вычислительных машин рассматриваемого кластера нейрокомпьютеров, очень важен параметр разрядности данных, поскольку за один процессорный такт на одном физическом вычислительном ядре нейрокомпьютера может выполниться до 64 операций, что существенно влияет на процесс распараллеливания вычислений в кластере, и что, в свою очередь, меняет процесс обхода графа потока управления.

В связи с этим граф потока управления программы будет представлен следующим образом $Gpu = (V, E, start, stop, BitVal)$, где $BitVal$ – множество значений разрядности результатов на каждом шаге выполнения программы при обходе графа потока управления.

Для разбиения программы на подпрограммы с позиции теоретических основ компиляции и построения компиляторов, необходимо рассмотреть программу во внутренней форме представления программ.

Внутренняя форма представления программ является результатом работы синтаксического анализатора. Различают следующие виды внутренних форм представления программ: постфиксная польская запись, префиксная польская запись, синтаксическое дерево, тетрады. Наиболее распространенным способом представления программ во внутренней форме является постфиксная польская форма [10, 11].

В рассматриваемой работе в качестве внутренней формы представления программ принята постфиксная польская форма записи. Польская форма записи традиционно применяется для записи математических и логических выражений. Для записи всей программы в польской форме необходимо применить модифицированый алгоритм перевода к польской записи.

Рассмотрим включение в польскую форму других, отличных от арифметических, операторов и конструкций:

*Безусловные переходы**:***

В польской форме безусловный переход – унарный оператор. Имеет следующий синтаксис:

<адрес перехода> $UT,

где $UT – знак операции, аналогично как в Польской записи арифметический выражений.

Условные переходы:

В польской форме – бинарный оператор со следующим синтаксисом:

<операнд 1>< адрес перехода > $CBZ |$CBlZ|$CBaZ|$CBleZ|$CBaeZ,

где *<операнд 1>* – значение арифметического выражения,*<адрес перехода >* – адрес перехода в таблице символов.

$CBZ – код оператора перехода по значению 0.

$CBlZ– код оператора перехода по значению меньше 0.

$CBaZ– код оператора перехода по значению больше 0.

$CBleZ– код оператора перехода по значению меньше или равно 0.

$CBaeZ– код оператора перехода по значению больше или равно 0.

Описание массива:

В польской форме – n-арный оператор со следующим синтаксисом:

<элемент массива 1>…<элемент массива i><адрес> $ARRAY,

где *<элемент массива i>* – адрес очередного элемента массива, *<адрес>* – в таблице символов, также хранит размерность массива, *$ARRAY*– оператор.

Обращение к элементу массива:

В польской форме – n-арный оператор со следующим синтаксисом:

<выражение 1> ... <выражение i><адрес> $SUBS,

где оператор *$SUBS*, используя *<адрес>* изтаблицы символов и индексные выражения, вычисляет адрес элемента массива.

Вызов подпрограммы с аргументами:

В польской форме – n-арный оператор со следующим синтаксисом:

<аргумент 1>…<аргумент i><имя функции> $CALL,

где *<аргумент i>*- аргументы функции.

Начало и конец программы:

В польской форме – безаперандные операторы со следующим синтаксисом:

$BEGIN $END [16]

После перевода программы в обратной Польской записи необходимо построить граф потока управления программы, который отобразит множество всех путей исполнения программы. Представление программы в расширенной обратной Польской записи хранится и передается в форме абстрактного типа данных класса стек. Будем называть такой стек – стек команд программы (*Stack*). Сформируем из стека команд программы абстрактное синтаксическое дерево (АСД) программы. Абстрактное синтаксическое дерево – конечное, помеченное, ориентированное дерево, в котором внутренние вершины помечены операторами языка программирования, а листья – соответствующими операндами. Листья самого нижнего слоя являются входными параметрами, а корень дерева результатом, который возвращает программа в результате своей работы. Ребрами АСД формируются между операторами и соответствующими им операндами, необходимыми для работы операторов. Для этого необходимо разработать алгоритмдля перевода программы из Польской формы в абстрактное синтаксическое дерево. Будем хранить в *R* текущий считываемый из *Stack*элемент, до тех пор пока он не пуст.

- Если *R* является идентификатором или константой, то его значение выбирается из стека, становится листом АСД и записывается в многосвязный список со ссылкой на родительский элемент в дереве, затем осуществляется считывание следующего элемента.
- Если *R* – бинарный оператор, то он применяется к двум следующим за ним элементам – операндам из стека, таким образом, формируется поддерево АСД, где родителем является бинарный оператор, а его детьми операнды. Затем осуществляется считывание следующего элемента.
- Если *R* – унарный оператор, то он применяется к верхнему последующему элементу из стека, таким образом, формируется поддерево АСД, где родителем является

унарный оператор, а его единственным наследником операнд. Затем осуществляется считывание следующего элемента.

- Иначе *R* *n*-арный оператор, он применяется ко всем nверхним последующим операндам из стека, таким образом, формируется поддерево АСД, где родителем является *n*-арный оператор, а его *n* наследников операнды. Затем осуществляется считывание следующего элемента.

Результатом работы предлагаемой программы, работающей в соответствии с представленным набором правил для перевода программы из Польской формы в абстрактное синтаксическое дерево является абстрактное синтаксическое дерево графа, хранящееся в памяти в виде многосвязного списка с пропусками по Р. Седжвику [1, 9].

Библиографический список

1. Вирт, Н. Алгоритмы и структуры данных / Н. Вирт ; пер. с англ. Ф. В. Ткачева. – М. : ДМК-Пресс, 2010. – 272 с.

2. Воеводин, В. В. Параллельные вычисления / В. В. Воеводин, Вл. В. Воеводин. – СПб. : БХВ-Петербург, 2002. – 608 с.

3. Гергель, В. П. Исследование алгоритмов планирования параллельных задач для кластерных вычислительных систем с помощью симулятора / В. П. Гергель, П. Н. Полежаев // Вестник Нижегородского университета им. Н. И. Лобачевского. – 2010. – № 5-1. – С. 201–208.

4. Грис, Д. Конструирование компиляторов для цифровых вычислительных маши / Д. Грис. – М. : Мир, 1975. – 544 с.

5. Романчук, В. А. Модель описания функционирования нейропроцессора как сложной параллельной системы / В. А. Романчук, В. В. Лукашенко // Автоматизация и управление в технических системах. – 2015. – № 2. – URL: auts.esrae.ru/14-266.

6. Романчук, В. А. Разработка математической модели реструктуризуемого под классы задач, виртуализируемого кластера GRID-системы на базе нейропроцессоров / В. А. Романчук, В. В. Лукашенко // Вестник Рязанского государственного университета им. С. А. Есенина. – 2014. – № 1 (42). – С. 177–182.

7. Ручкин, В. Н. Обобщенная модель вычислений кластера нейрокомпьютеров / В. Н. Ручкин, В. А. Романчук, В. В. Лукашенко // Вестник Рязанского государственного университета им. С. А. Есенина. – 2015. – № 2 (47). – С. 146–150.

8. Танненбаум, Э. Распределенные системы. Принципы и парадигмы / Э. Танненбаум, М. Ван Стен. – СПб. : Питер, 2003. – 877 с.

9. Тель, Ж. Введение в распределенные алгоритмы : пер. с англ. / Ж. Тель. – М. : МЦНМО, 2009. – 616 с.

10. Топорков, В. В. Модели распределенных вычислений / В. В. Топорков. – М. : ФИЗМАТЛИТ, 2004. – 320 с.

А. А. Симаков, А. С. Макурков, С. В. Шибанов

Пензенский государственный университет, Пенза, Россия

СРАВНИТЕЛЬНЫЙ АНАЛИЗ ИНСТРУМЕНТОВ ВИЗУАЛИЗАЦИИ ПОВЕДЕНИЯ ПРАВИЛ В АКТИВНЫХ БАЗАХ ДАННЫХ

Рассмотрены и проанализированы основные инструменты визуализации поведения правил в активных базах данных. Выявлены их достоинства и недостатки.

Правила в активной базе данных выполняются «за кулисами», это означает, что пользователи системы не увидят, когда правила выполнятся. Предположим, что набор правил содержит некоторые правила, исполнение которых приведет к нежелательному состоянию системы. Поскольку выполнение правила происходит «за кулисами», сбой произойдет

без каких-либо предшествующих предупреждений. Также будет трудно понять, что пошло не так после этого.

Следовательно, необходим инструмент визуализации поведения правил для облегчения процесса разработки и технического обслуживания активных баз данных. Без такого инструмента будет очень сложно отследить сложные взаимодействия между правилами, которые в итоге могут привести к серьезной неисправности. Инструментарий визуализации мог бы, если не решить эту проблему, по крайней мере, предупредить о сбое и, если этот сбой произойдет, упростить анализ ситуации [1].

Поскольку исследования в области визуализации активных правил не имеют массовый характер, найти готовые работы и решения очень трудно. Наиболее серьезная работа в этом направление проводилась в рамках проектов DEAR и Adela.

DEAR-отладчик для активных правил в объектно-ориентированном контексте. Он был реализован на основе менеджера правил EXACT, поддерживающего ECA (event – condition – action) правила в объектно-ориентированной системе баз данных ADAM.

DEAR отслеживает не только правила, но и события. События трассировки дают важные подсказки пользователю – цикл связанных событий позволяет пользователю узнать не только о том, какие правила были запущены, но и то, что привело к запуску этих правил. И наоборот, когда возникает событие, позволяет обнаружить контекст, в котором возникло это событие [2].

Контекст может быть интерпретирован как набор правил коллизии. Недостаток с отображением последовательности при отладке правил заключается в том, что отладчик будет игнорировать тот факт, что правило должно конфликтовать с другими правилами во время их исполнения.

Дерево отладки может легко «разрастись» в процессе анализа, поэтому DEAR предлагает несколько способов его отображения:

– отслеживание событий (*spy_event*) позволяет отслеживать определенные события в отладке;

– отслеживание правил (*spy_rule*) позволяет отслеживать определенные правила в отладке;

– отслеживание данных (*data_spy*) позволяет отслеживать определенные данные в отладке, т.е. дерево будет генерироваться только в случае обновления контролируемых данных.

Попытка решить проблемы и разрешить недостатки DEAR была произведена при разработке системы Adela. Рассмотрим ее особенности и структуру подробнее.

Adela реализуется как прототип на языке программирования C++, используя анимационную библиотеку Goofy. Goofy построена на основе другой библиотеки Polka, разработанной Джоном Т. Стаско в Колледже вычислительной техники Института технологии, Атланта. Goofy позволяет осуществлять косвенный доступ к Polka через интерфейс высокого уровня.

Визуализация, реализованная в Adela, предусматривает два окна анимации: окно трассировки событий/правил и окно трассировки данных [3].

Окно трассировки событий/правил, представленное на рис. 1, визуализирует последовательность событий и правил, а также состояние выполнения каждого правила при его активации.

Окно данных, на рис. 2, визуализирует действие правил. Поскольку атрибуты и экземпляры игнорируются в прототипе, окно трассировки данных показывает только операции правил по типам объектов. «Экземпляры», на самом деле, не являются экземплярами типов объектов: их следует рассматривать как множества экземпляров.

Adela представляет пользователю дерево трассировки событий/правил, что делает возможным отображение контекста, в котором запускаются правила. Дерево также может отображать состояние выполнения правил, т.е. когда правило было запущено, когда были проверены условия и когда выполнено действие правил. Также важным инструментом системы Adela является отображение трассировки данных. Цель трассировки данных заключается в том, чтобы отображать, что происходит с экземплярами данных, когда ими манипулируют активные правила.

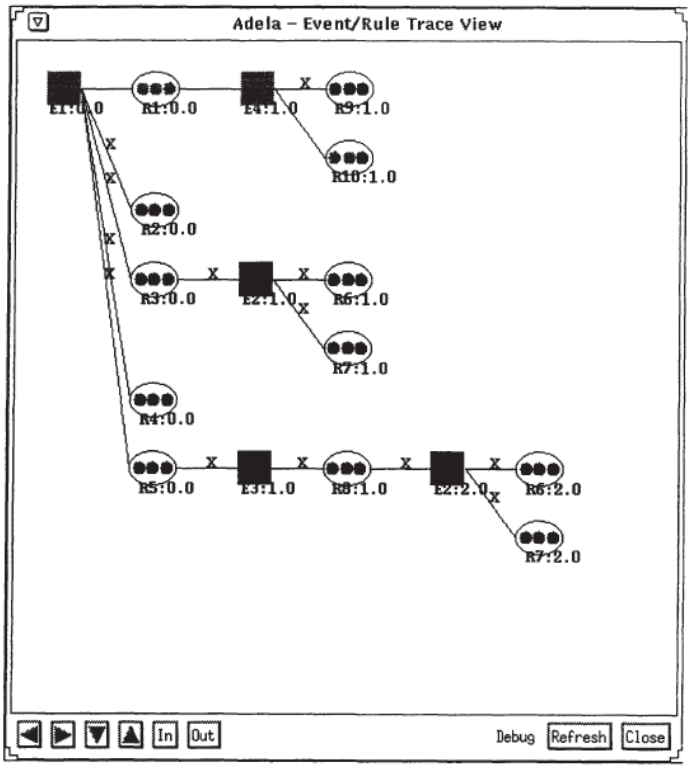


Рис. 1. Окно трассировки событий/правил

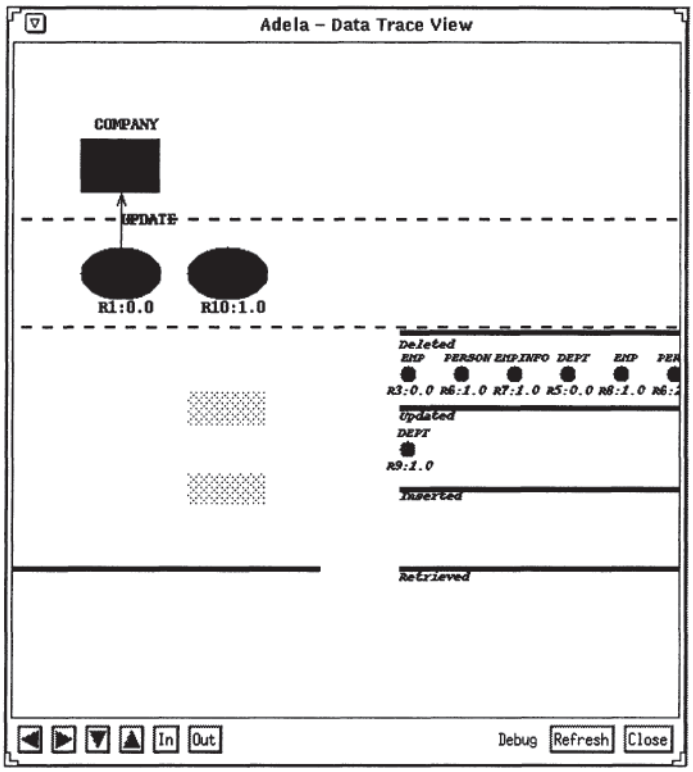


Рис. 2. Окно трассировки данных

Таким образом, визуализация поведения активных правил в системах баз данных является важной областью исследований. Однако, остается достаточно много неразрешенных вопросов, например, визуализация поведения правил в режиме реального времени.

Библиографический список

1. Fors, T. Visualization of rule behaviour in active databases / T. Fors. – Sweden : University of Skovde, 1995. – P. 182–196.
2. Diaz, Jaime A. DEAR: a debugger for active rules in an object-oriented context, in Rules in Database Systems / A. Diaz, Jaime, N. W. Paton. – Springer-Verlag, 1993.
3. Fors, T. ADELA: animated debugging and explanation of active database rules, MSc. dissertation, department of computer science / T. Fors. – Sweden University of Skovd, 1994.

О. О. Очередько, П. Н. Полежаев

Оренбургский государственный университет, Оренбург, Россия

ВОЗМОЖНОСТИ ДЛЯ РАСПАРАЛЛЕЛИВАНИЯ НЕЙРОННЫХ СЕТЕЙ

В настоящее время нейронные сети используются для решения многих задач. В статье описываются основные направления применения нейронных сетей в зависимости от класса решаемых задач с указанием типов используемых сетей. Также подробно приводятся возможные уровни распараллеливания нейронных сетей. В заключении дается описание технологий для реализации параллельных алгоритмов искусственных нейронных сетей.

Нейронные сети эффективны для решения многих проблем, в которых математическое моделирование весьма затруднено. Они используются для решения задач распознавания речи, информационного поиска с учетом контекстной или персонифицированной информации, в задачах распознавания и автоматической обработки изображений, автоматического управления автомобилями, беспилотными летательными аппаратами и т.д.

Существует четыре основных направления применения нейронных сетей, которые имеют большое значение:

1. Классификация. В математическом смысле это связано с делением n-мерного пространства на различные области и с указанием точки, которая принадлежит одной из этих областей. Эта идея используется при решении многих задач по распознаванию образов. Каждый шаблон преобразуется в многомерную точку и классифицируется в определенную группу, каждая из которых представляет собой уже известный шаблон. Тип используемой сети: нейронная сеть прямого распространения.

2. Прогнозирование. Нейронную сеть можно обучить для прогнозирования, которое будет выполняться и зависеть от конкретных заданных значений. Если есть сеть, которая хорошо подходит для моделирования известной последовательности значений, то ее можно использовать для прогнозирования будущих результатов. Очевидным примером является прогнозирование на фондовом рынке. Тип используемой сети: нейронная сеть прямого распространения.

3. Кластеризация. Существуют данные, анализ которых настолько сложный, что нет очевидного способа классифицировать их по разным категориям без предварительной кластеризации. Этот метод полезен для интеллектуального анализа данных как для коммерческих, так и для научных целей. Типы используемой сети: простые конкурентные сети, сеть адаптивного резонанса, самоорганизующаяся карта Кохонена.

4. Ассоциация. Нейронную сеть можно обучить «запоминать» несколько шаблонов таким образом, что при вводе искаженной версии определенного шаблона сеть связывает ее с наиболее подходящим шаблоном и возвращает его исходную версию. Это полезно для восстановления искаженных и зашумленных данных. Тип используемой сети: нейронная сеть Хопфилда.

По мере увеличения масштаба нейронной сети требуется большой объем данных, что приводит к экспоненциальному увеличению времени обучения и работы нейронной сети. Поэтому крайне важно, чтобы модели нейронных сетей работали быстро и генерировали результаты в реальном времени.

Параллельное выполнение нейронных сетей необходимо из-за того, что обучение каждого узла в крупных сетях может продолжаться очень длительное время. Однако нейронные сети используют сетевые топологии, которые легко можно распараллелить в тех случаях, когда вычисление каждого узла, как правило, не зависит от других узлов [1].

Модель нейронной сети состоит из нескольких параллельных структур, которые используются для улучшения эффективности реализации на параллельных архитектурах. Существуют несколько уровней распараллеливания искусственных нейронных сетей [2]:

- Уровень фазы обучения. При решении задач посредством нейронной сети обычно проводят большое количество экспериментов для определения начальных оптимальных параметров, таких как количество нейронов в различных слоях, параметр скорости обучения и т.д. [3]. При использовании распараллеливания на уровне фазы обучения различные конфигурации нейронной сети могут исследоваться одновременно. Единственным отличием между двумя фазами будет параметр скорости обучения. На данном уровне легко достигается линейный прирост быстродействия, так как исчезает необходимость межпроцессорного взаимодействия.
- Уровень обучающей выборки. Размеры обучающей выборки, необходимой для решения некоторой задачи, часто могут быть достаточно большими. И обучающие вектора будут направлены в сеть по одному в однопоточной системе, тогда, как в параллельной системе эти вектора можно разделить между процессами. Каждому процессу необходима полная копия нейронной сети и таким образом, несколько копий сети будут обучаться одновременно на разных обучающих выборках [3].
- Уровень слоя. В многослойных моделях нейронной сети обучающие вектора проходят через сеть по принципу конвейера, таким образом несколько обучающих векторов могут находиться в сети одновременно. Однако в однослойных моделях входной слой не производит никаких вычислений, а только распределяет входные вектора на нейроны основного слоя, то при распараллеливании на данном уровне в таких моделях прироста производительности достичь нельзя [3].
- Уровень нейрона. Распараллеливание на данном уровне является наиболее очевидным среди всех уровней распараллеливания искусственных нейронных сетей [3]. Распараллеливание на нейронном уровне заключается в разделении нейронов между процессорами и последующем параллельном вычислении. Одному процессору относится один или несколько нейронов сети. Распараллеливание на уровне нейрона является наиболее популярном методом во многих реализациях, так как не зависит от используемой модели нейронной сети.
- Уровень весов. Вычисления в пределах нейрона также могут быть разделены среди нескольких процессоров. Данное распараллеливание является мелкомодульным, и оно представлено в первую очередь в аппаратных реализациях [3].

Все уровни распараллеливания могут использоваться одновременно в нейронной сети, но это не всегда приводит к оптимальному результату [4]. Так, многоуровневая структура многих сетей с прямой связью и их методы обучения с обратным распространением ошибки сильно ограничивают ускорение нейронных вычислений. Для них будет более практичным использование распараллеливания на уровне обучающей выборки или

уровне слоя. Для однослойных сетей будет более эффективным использование распараллеливания на уровне нейрона (для SOM или рекуррентных сетей), так как этот метод не зависит от используемой модели нейронной сети [5].

В настоящее время существует несколько технологий для реализации параллельных алгоритмов искусственных нейронных сетей:

- Суперкомпьютеры. Суперкомпьютеры являются одним из самых распространенных решений для параллельного выполнения различных моделей нейронной сети [6]. Современные системы включают в себя полный набор программ, оптимизированных для глубокого обучения, что позволяет исследователям и ученым в области данных быстро и легко обучать глубокие нейронные сети. Помимо этого, они включают в себя оптимизированные версии нескольких широко используемых библиотек глубокого обучения. Современные модели могут также дополнительно предоставлять доступ к облачным инструментам управления, обновлениям программного обеспечения и банку приложений-контейнеров.
- Нейрочипы. Нейрочипы являются менее распространенным вариантом использования из-за своей дороговизны. В основном их делают под заказ и не производят в массовом масштабе, что делает их невостребованными среди обычных пользователей.
- Многопроцессорные системы. Многопроцессорные системы чаще всего используют в крупных научных центрах, а использование их для задач общего типа становится невозможным из-за узкой специализации большинства решений и проблем с возможностью их масштабирования.
- GPGPU (General-purpose computing on graphics processing units). GPGPU – это графический процессор, который выполняет неспециализированные вычисления, которые обычно выполняются центральным процессором. Примеры реализации GPGPU: CUDA, DirectCompute, OpenCL, AMD FireStream, OpenACC, C++ AMP.

Таким образом все перечисленные технологии успешно используются для распараллеливания нейронных сетей. Наиболее распространенным вариантами являются суперкомпьютеры и GPGPU. Менее распространенными решениями являются нейрочипы и многопроцессорные системы.

Библиографический список

1. Udo Seiffert, Artificial Neural Networks on Massively Parallel Computer Hardware // European Symposium on Artificial Neural Networks Bruges (Belgium) – 2002. – P. 319–330.

2. Nordstrom, T. Designing parallel computers for self-organizing maps / T. Nordstrom // Forth Swedish Workshop on Computer System Architecture. – Linkoping, 1992.

3. Боголюбов, Д. П. Реализация алгоритма обучения самоорганизующихся карт Кохонена на графических процессорах / Д. П. Боголюбов, А. А. Чанкин, К. В. Стемиковская // Промышленные АСУ и контроллеры. – 2012. – № 10. – С. 30–35.

4. Serbedzija, N. B. Simulating artificial neural networks on parallel architectures / N. B. Serbedzija // Computer. – 1996. – P. 56–63.

5. Yoshiyasu Takefuji. Neural Network Parallel Computing / Yoshiyasu Takefuji. – USA, Kluwer Academic Publishers, 1992. – 210 c.

6. D. Valencia, A. Parallel Processing of High-Dimensional Remote Sensing Images Using Cluster Computer Architectures / D. Valencia, A. Plaza, P. Martinez, J. Plaza // International Journal of Computers and their Applications, USA, NC, Cary, March. – 2007. – Vol. 14, № 1. – P. 23–34.

7. Суперкомпьютер NVIDIA DGX-1. – URL: http://forsite-company.ru/services_and_solu-tions/dgx/ (дата обращения: 29.06.17).

М. М. Черепащук

Пензенский государственный университет, Пенза, Россия

РАЗРАБОТКА СИСТЕМЫ АВТОМАТИЧЕСКОЙ ГЕНЕРАЦИИ ИГРОВЫХ СИТУАЦИЙ НА ПРИМЕРЕ RPG-ИГР

Посвящена особенностям проектирования и реализации компьютерной игры на языке C# с применением механизмов процедурной генерации.

Компьютерные игры или видеоигры – вид программного обеспечения, предназначенный для организации игрового процесса, связи с партнерами по игре, или сама выступающая в качестве партнера [1].

Компьютерные игры используются в различных целях: некоторые видят в видеоиграх лишь средство развлечения или отдыха, другие применяют их для развития интеллекта и логического мышления, а для некоторых, компьютерные игры становятся настолько важной частью жизни, что заменяют реальный мир, становясь наркотиком. Также видеоигры разделяются по жанрам: от обычных «стрелялок» от первого лица до сложных симуляторов.

С точки зрения разработки игровых событий существует 2 основных подхода:

1) создание игровых событий вручную;
2) создание систем процедурной генерации событий.

При создании игровых событий вручную, программист самостоятельно задает начальные условия для каждого такого события, а также не ограничивает себя в разнообразии типов игровых событий и условий их выполнения. Например, сложные головоломки с использованием физических законов.

Так как такой способ требует много времени и трудовых ресурсов, то в основном используется в крупнобюджетных игровых проектах AAA – класса, целью которых, в подавляющем большинстве случаев, является повествование, а не игровой процесс.

В случае с использованием систем процедурной генерации события должны создаваться по определенным условиям, а также привязываются к определенным тематикам. Например, генерация случайных имен неигровых персонажей в играх, расположение некоторых игровых объектов в мире, генерация лабиринтов, генерация миров.

Вся сложность данного способа зависит лишь от функционала генератора, поэтому генерации игровых ситуаций чаще всего применяется в независимых проектах, так как удешевляют и ускоряют разработку.

На сегодняшний день все больше разработчиков ориентируются на использование систем генерации. В своем цикле статей [1] о процедурной генерации разработчик Марк Джонсон описывает большое число возможных способов применения систем генерации. Эти статьи и послужили толчком к определению темы выпускной квалификационной работы.

Исходя из поставленной цели, были сформулированы следующие задачи:

1. Исследование существующих аналогов, и выявление их достоинств и недостатков.
2. Проектирование основных игровых подсистем.
3. Разработка игры, для демонстрации спроектированных игровых подсистем.

На этапе анализа были определены программы-аналоги и их недостатки, которые можно было бы исправить при разработке собственной игровой системы.

Для определения состава игровой системы необходимо было ответить на следующие вопросы:

1. Какие действующие лица будет взаимодействовать с игровой системой?
2. Какие функции будут доступны этим лицам?

Чтобы ответить на эти вопросы была построена диаграмма прецедентов (рис. 1).

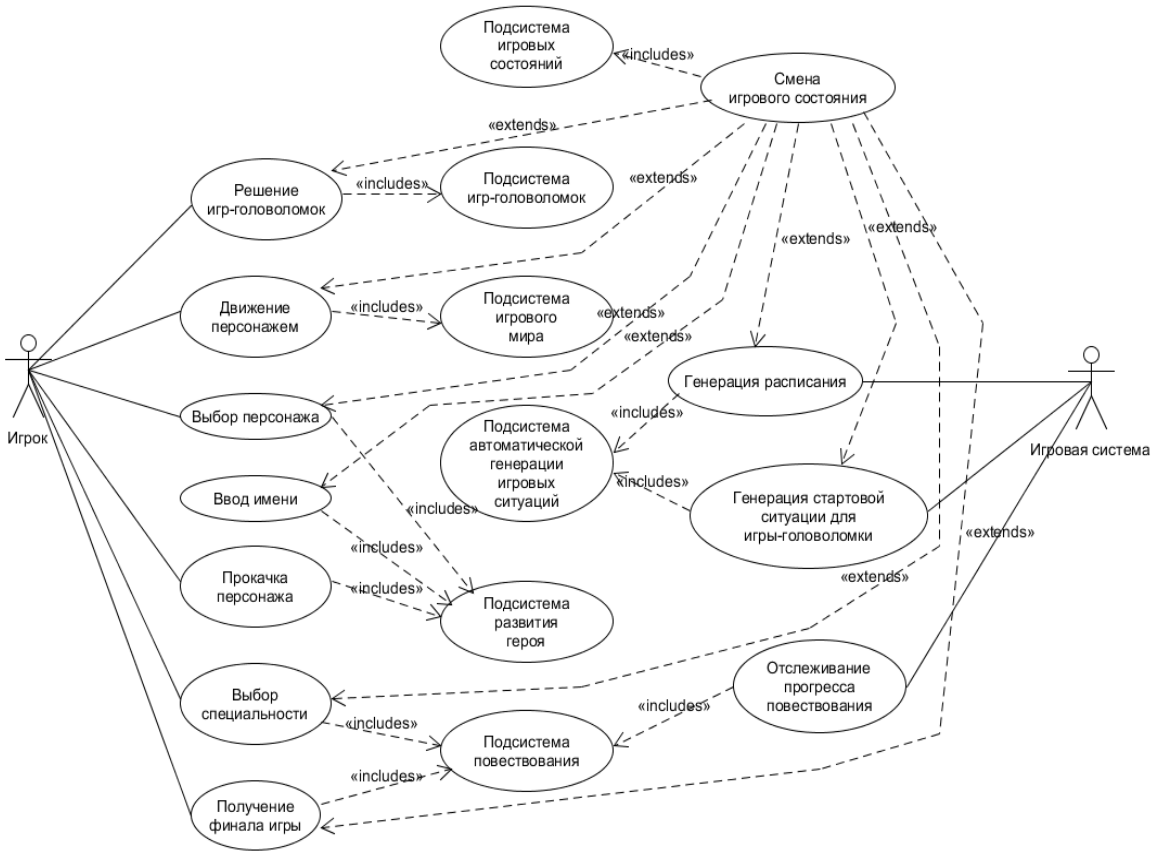


Рис. 1. Диаграмма прецедентов

На основе данных из диаграммы прецедентов был определен набор главных игровых подсистем:

1. Подсистема игровых состояний – предназначена для управления игровыми состояниями и переходами между ними. В результате проектирования данной подсистемы были определены основные игровые состояния.

2. Подсистема игрового мира – обеспечивает существование открытого мира игры, управление персонажем внутри него, перемещением по локациям и взаимодействием с объектами внутри игрового мира. При проектировании этой подсистемы были определены локации, способы перехода между ними, карта игрового мира и возможности управления персонажем.

3. Подсистема повествования – обеспечивает сюжетную составляющую игры. При проектировании были сформулированы основные игровые этапы и правила игры и условия прохождения.

4. Подсистема развития героя – предоставляет возможность выбора имени, пола, специальности и обеспечивает управление развитием героя. В результате проектирования были определены основные характеристики героя и способы его развития.

5. Подсистема игр-головоломок – обеспечивает управление играми-головоломками. В данной подсистеме были определены основные игры-головоломки и их правила.

6. Подсистема автоматической генерации игровых ситуаций – предназначена для формирования расписания и генерации стартовых позиций игр-головоломок. Данная подсистема реализует процедурную генерацию игровых ситуаций. В результате проектирования были определены правила, по которым генерировались игры головоломки.

При проектировании игровой системы также были определены основные входные и выходные параметры подсистем, с помощью которых они могли взаимодействовать между собой. Таким образом была построена структурная диаграмма взаимодействия подсистем (рис. 2).

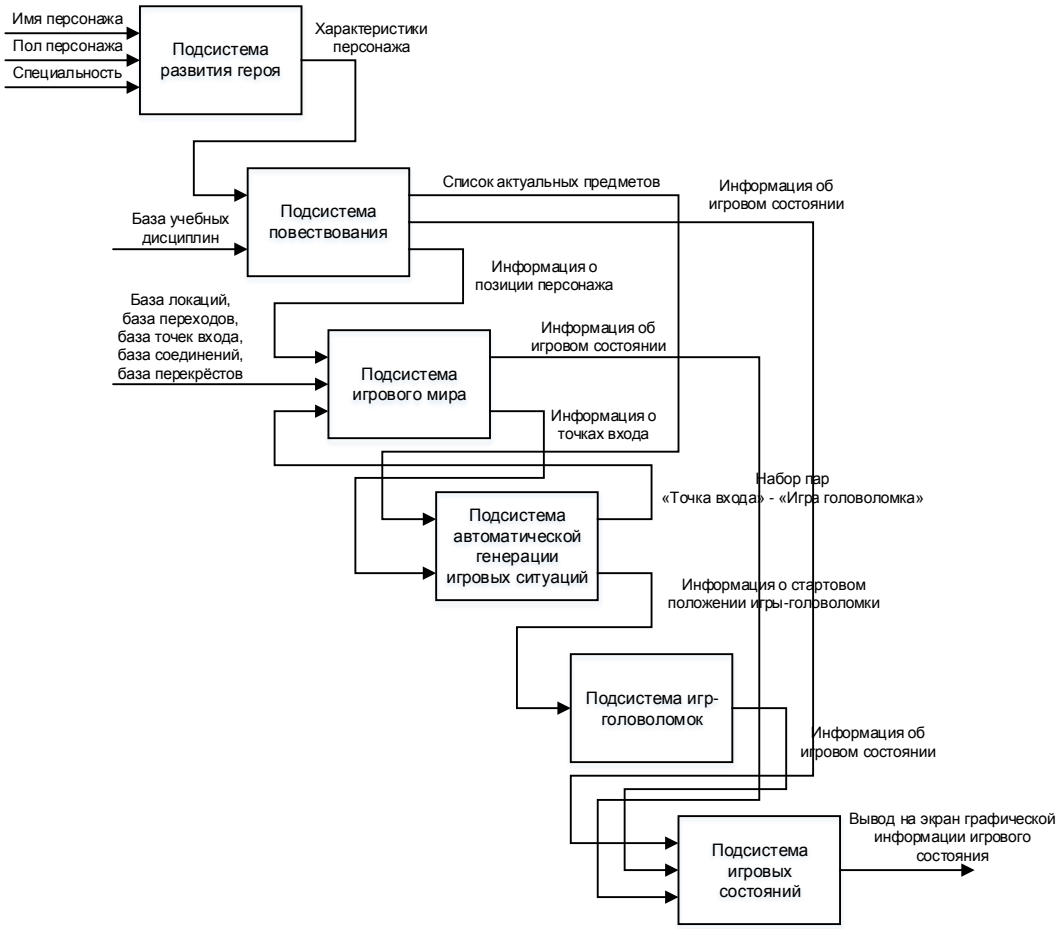


Рис. 2. Схема взаимодействия подсистем

Программирование игры осуществлялась на языке C# с использованием фреймворка MonoGame. Язык C# был выбран по причине простоты в изучении, наличием богатой стандартной библиотеки классов и высокой производительностью написанных программ, а наличие фреймворка MonoGame позволило упростить реализацию базовых игровых компонентов. В результате программной реализации игровой системы были написано 53 класса, реализующие все игровые подсистемы.

В результате, всю проделанную работу можно разделить на 3 части: анализ области, проектирование игровой системы и ее программная реализация. При программировании основной упор делался на 2 критерия:

1. Расширяемость системы – возможность добавления новых компонентов, не требующего значительного изменения уже имеющихся компонентов.

2. Оптимальное использование аппаратных ресурсов компьютера для эффективной работы игровой системы.

Библиографический список

1. Компьютерная игра. – URL: https://ru.wikipedia.org/wiki/ (дата обращения: 10.09.2017).

2. Mark Jonson, Generation Next. – URL: https://www.rockpapershotgun.com/2016/07/22/future-of-procedural-generation-1/ (дата обращения: 10.09.2017).

3. Троелсен, Э. Язык программирования C# 5.0 и платформа .NET 4.5 / Э. Троелсен. – М. : Вильямс, 2013. – 1311 с.

4. Рихтер, Дж. CLR via C#. Программирование на платформе Microsoft .NET Framework 4.5 на языке C# / Дж. Рихтер. – 4-е изд. – СПб. : Питер, 2013. – 896 с.

5. MonoGame Documentation. – URL: http://www.monogame.net/documentation/?page=main (дата обращения: 10.09.2017).

А. В. Жохов

Пензенский государственный университет, Пенза, Россия

НАПРАВЛЕНИЯ РАЗВИТИЯ СИСТЕМЫ ПРОГНОЗИРОВАНИЯ И МОНИТОРИНГА ПОТРЕБНОСТЕЙ РЫНКА ТРУДА В ПРОФЕССИОНАЛЬНЫХ КАДРАХ

Посвящена сравнительному анализу систем мониторинга и прогнозирования и выявлению направлений развития системы, разработанной в Пензенском государственном университете.

На сегодняшний день наиболее востребованными в области решения задач согласования образовательного рынка и рынка труда являются системы, специализирующиеся на информационной поддержке согласованного выпуска специалистов с потребностями рынка труда. В [1] был проведен сравнительный анализ таких систем. Полученные результаты показали, что необходимо разработать систему мониторинга и прогнозирования, которая удовлетворяла бы всем критериям, приведенным в [1], и в которой имелась бы возможность непосредственного взаимодействия с работодателями и образовательными учреждениями. Такая система была разработана в Пензенском государственном университете аспирантом А.В. Лушниковым, под руководством профессора А. М. Бершадского.

Существует еще одна система, которая не была рассмотрена в [1], но определенно заслуживает внимания, так как может составить конкуренцию разработанной в ПГУ системе.

Исходя из выше сказанного, возникает необходимость в проведении сравнительного анализа предложенных систем.

1. Система мониторинга и прогнозирования потребностей рынка труда в кадрах различной специализации и квалификации (ПГУ).

Для повышения эффективности взаимодействия системы образования региона и рынка труда в Пензенском государственном университете была разработана система мониторинга и прогнозирования потребностей рынка труда в профессиональных кадрах. Эффективность достигается за счет автоматизации отдельных этапов технологического процесса прогнозирования и мониторинга, а именно:

- создание и ведение информационных фондов мониторинга потребностей рынка труда в кадрах по уровням, направлениям и объемам подготовки;
- создание и ведение информационных фондов мониторинга систем общего и профессионального образования;
- организация информационного взаимодействия с участниками мониторинга для обеспечения сбора необходимых исходных данных;
- информационное обслуживание пользователей в среде корпоративной системы;
- проведение расчетов по выявлению, оценке, среднесрочному и долгосрочному прогнозированию тенденций развития регионального рынка труда с учетом социально-экономических факторов;
- прогнозирование выпуска из системы профессионального образования;
- формирование проектов контрольных цифр приема на подготовку кадров с профессиональным образованием;
- формирование и оформление результатов мониторинга и прогнозирования;
- обеспечение доступа граждан к обобщенной информации, основанных на результатах мониторинга и прогнозирования.

2. Экспертно-аналитическая система прогнозирования потребностей в профессиональных кадрах (Информационные Бизнес-Системы – IBS).

Для поддержки формирования контрольных цифр приема компанией IBS по заказу Минобрнауки России была разработана Экспертно-аналитическая система прогнозирова-

ния потребностей в профессиональном кадровом обеспечении социально-экономического развития Российской Федерации. IBS – одна из ведущих российских IT-компаний, крупнейший разработчик аналитических систем и решений для сферы образования России.

Данная система позволяет проводить экспертизу и анализ потребностей экономики РФ и субъектов РФ в специалистах разного профиля, а также анализировать соответствие структуры подготовки специалистов прогнозу потребности в кадрах различных отраслей экономики.

Ядром системы являются созданные экспертами IBS 83 региональных динамических модели социально-экономического развития, которые позволяют определить структуру потребности экономики в кадрах на среднесрочную и долгосрочную перспективу. Эти модели, в свою очередь, агрегируются на федеральном уровне. В результате моделирование развития страны происходит в единой модели социально-экономического развития РФ. Такой подход позволяет рассчитать будущую потребность экономики в кадрах различного профессионального профиля по каждому субъекту отдельно и по Российской Федерации в целом. Эти расчетные данные затем становятся основой для формирования утверждаемых Минобрнауки России контрольных цифр приема в вузы по всей стране в текущем году.

В экспертно-аналитической системе реализованы механизмы сценарного анализа. Система позволяет анализировать различные сценарии развития экономики – основной сценарий (базовый), оптимистический, консервативный и др., и таким образом вырабатывать наиболее актуальные решения о необходимости развития системы образования в стране. Математические модели и методики, заложенные в основу системы, прошли многоэтапную экспертизу в научном сообществе и в Минобрнауки России.

С технологической точки зрения, система, созданная специалистами IBS, представляет собой уникальную программную разработку. Модели реализованы в среде Any Logic. Система предоставляет специалистам Министерства возможность осуществить доступ к исходным статистическим и прогнозным данным, анализ и сценарное моделирование, процесс согласования прогнозных цифр через веб-интерфейс на специализированном портале [3].

С учетом сказанного в табл. 1 представлены результаты сравнительного анализа.

Таблица 1

Сравнительный анализ систем мониторинга

Возможности	Системы	
	Система мониторинга и прогнозирования потребностей рынка труда в кадрах различной специализации и квалификации	Экспертно-аналитическая система прогнозирования потребностей в профессиональных кадрах
Расчет КЦП	Да	Да
Прогнозирование возможностей системы образования	Да	Нет
Прогнозирование возможностей экономики	Да	Нет
Анализ спроса и предложения на рынке труда	Да	Нет
Способ представления результатов	Табличный(HTML, Excel, PDF); Графически (Гистограмма, график, лепестковая диаграмма, секторная диаграмма)	Таблицы, Графики, Гистограммы
Непосредственное взаимодействие с участниками мониторинга	С образовательными учреждениями, с работодателями	С работодателями
Анализируемые уровни образования	НПО, СПО, ВПО (специалитет)	ВПО (учитывая бакалавриат, магистратуру, специалитет; без учета профилей)

Подробно изучив организацию работы каждой из предложенных систем можно сказать, что каждая обладает своими достоинствами и недостатками. Кроме того, сравнительный анализ позволил выделить направления, в которых можно продолжать развивать систему, разработанную в ПГУ, что позволит поддерживать ее востребованной и конкурентоспособной:

- расширение анализируемых уровней образований согласно современным стандартам образования;
- расширение возможностей представления статистических данных;
- разработка модуля автоматизированной подготовки документов;
- организация интерфейса.

Библиографический список

1. Лушников, А. В. Методики и алгоритмы принятия решений при подготовке профессиональных кадров для регионального рынка труда : дис. … канд. техн. наук / Лушников А. В. – Пенза, 2014. – С. 23–32.

2. Центр бюджетного мониторинга. – URL: http://openbudgetrf.ru/web-resources/образование-в-регионах-российской-фе/

3. Экспертно-аналитическая система прогнозирования. – URL: https://www.ibs.ru/projects/minobrnauki-rf-ekspertno-analiticheskaya-sistema-prognozirovaniya/?sphrase_id=19290.

4. Система прогнозирования кадровых потребностей РФ. – URL: http://profprognoz.ru/question.

Yu. I. Serikova

Penza State University, Penza, Russia

COMPARISON OF PREDICTORS WITH LINEAR COMPUTATIONAL COMPLEXITY

A general scheme for the synthesis of the analog and quantum forms of an oracle is given. The oracle predicts the correlation coefficients for thirty-two experiments. As a result, our study shows that the oracle calculation using the continuous form allows one to decrease the standard deviation of the computational error by 11,6 %, which leads to a 30 % increase in the number of experiments; quantum – 85 %. Thus, we obtain that quantum oracles are much more effective than the analog forms of the oracle.

Continuous computations of lower-order statistical moments and correlation coefficients

Biometry has mainly statistical character. The more data are obtained in the reference sample, the more unique and reliable the system is. Actually, the data is always insufficient. Statistical multi-variable estimates are significantly complicated for computation of human biometric data. However, it is noteworthy that a double-triple decrease of the test sample size can result in the double-triple decrease of the amount of time needed to test biometric data and the whole system.

The problem of biometric technology testing in the case of small sample sizes is standardized. However, this standard is valid for the means of neural-network biometry in the case of the neural networks being trained by standard [1].

If one stayed within the biometric framework then the problem would be reduced to the decrease of computational errors of the biometric parameters. There are two kinds of such errors: the computational error of the expectation (ΔE) and the computational error of the standard

deviation ($\Delta\sigma$). We consider the samples of the small size – n. In our case the computational error (ΔE) is much smaller than the error $\Delta\sigma$. As a result the error accumulation occurs:

$$\begin{cases} \Delta E(\nu_1) = f(\nu_1, n) \\ \Delta\sigma(\nu_1) = f(\nu_1, \Delta E(\nu_1), n) \end{cases} \quad (1)$$

The computational error of correlation coefficients is even higher since it is now affected by the four errors of the lower-order statistical moments:

$$\Delta r(\nu_1, \nu_2) = f(\nu_1, \nu_2, \Delta E(\nu_1), \Delta E(\nu_2), \Delta\sigma(\nu_1), \Delta\sigma(\nu_2), n) \quad (2)$$

Computational instability increases as the number of variables does. As a result we need to compensate the computational errors of intermediate data (regularization procedures). As it is shown in works [2, 3] it is possible to decrease the methodological and random components of the computational errors by 1,5–30 % depending on the size – n of the test sample. All the computations in works [2–4] are carried out in the continuous form.

Continuous correction of pair correlation coefficients

Usually the correlation coefficient is calculated by the Pearson formula:

$$r(x_1, x_2) = \sum_{i=1}^{32} \frac{(E(x_1) - x_{1,i}) \cdot (E(x_2) - x_{2,i})}{\sigma(x_1) \cdot \sigma(x_2)} \quad (3)$$

It gives a significant error Δr as a result. The maximum of the error arises when the correlation coefficients which are close to zero are calculated ($\max(\sigma(r))$ at r = 0). Thus, the most interesting case is computational error correction for weakly correlated data. The result of a similar linear correction for "fractal correlation functionals" [2] allows one to decrease the computational error by 11.1%.

A data correction mathematical molecule could be obtained by simple averaging:

$$\tilde{r}(x_1, x_2) = \frac{r(x_1, x_2) + R(x_1, x_2)}{2} \quad (4)$$

In this case the correlation between quantum-continuous and continuous forms of calculation is high and positive. Averaging of data over the sample of thirty-two experiments leads to an 11?6 % decrease of the standard deviation ($\sigma_{32}(\tilde{r})$) in comparison with the standard deviation of the Pearson correlation ($\sigma_{32}(r)$). From the statistical point of view, it means that the sample of thirty-two experiments increases surprisingly to thirty-nine experiments. We have obtained a significant (about 30 %) increase of the experiments due to the fact that we have additionally applied the second form (4) of correlation functionals instead of the classical form (3) of correlation functionals. Two observers of the correlation coefficient seem to be not fully dependent; their errors seem to be different:

$$\Delta r_{32}(x_1, x_2) \neq \Delta R_{32}(x_1, x_2) \quad (5)$$

Quantum corrector of the correlation coefficient by spectrum

It is technically unfeasible to implement the solution of the Schrödinger equation in the case of large amount of electrons; it requires high computational performance [5]. This is precisely the main problem of quantum mechanics: the performance of modern computers is insufficient if we try to calculate the wave functions for thirty-two electrons. Everything drastically changes for the mathematical correlation molecule. A significant advantage of the mathematical molecule consists in the fact that nine lines of high-level language code are enough to simulate it. An example of the program implementation of the mathematical molecule is given in Figure 1. It should be emphasized that in the case of sixteen experiments the spectrum of outcome states for the mathematical correlation molecule can contain up to sixteen spectral lines (see figure 1 for the program implementation).

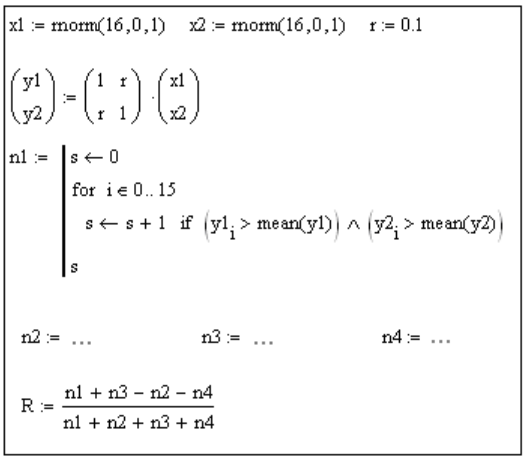


Figure 1. The program implementation of the mathematical molecule for a sample of 16 experiments

Figure 2 (data from [6]) represents two spectra obtained at r =0 and at r = 0.5 for the sample of nine hundred ninety-nine implementations of sixteen experiments in each. The left hand side of Figure 2 shows that the tenth spectral line with the ordinate r = 0.0 has a maximum of the occurrence probability; moreover, the spectrum is symmetric.

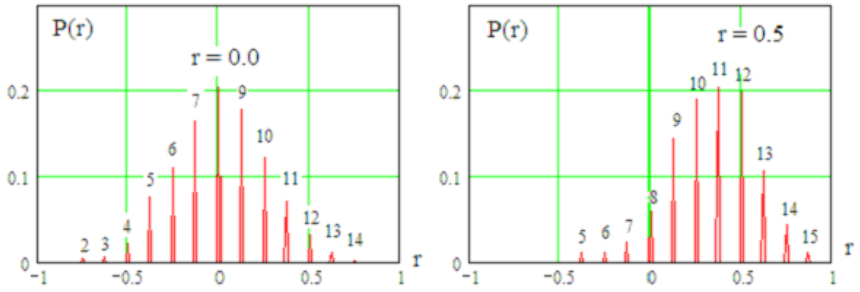


Figure 2. Examples of the spectra of the correlation coefficients for the samples of 16 experiments

It means that the correlation coefficient could be calculated by spectrum as an expectation of all the spectral lines detected:

$$\tilde{R} = \frac{1}{15}\sum_{i=1}^{15} P(r_i) = \frac{1}{15}\sum_{i=1}^{15} P(-1 + \frac{1}{16+1} + \frac{i}{16+1}) = E(P(r_i)). \quad (6)$$

Limiting values $P(r=-1)$ and $P(r=+1)$ do not contribute to formula (6), since their spectral components are totally determined and the probability of their occurrence in the spectrum tends to 0. It is obvious that a sample of about nine hundred ninety-nine implementations of sixteen experiments in each is required to observe the spectral lines. Such a sample could be obtained by using one sample of thirty-two experiments. One could do it by random extraction of multiple smaller samples of sixteen experiments from it. It is possible to obtain six hundred and one million of series of experiments without repetition. It is an upper bound and it is not necessary to test such a large amount of series. Close samples are low informative and it is not necessary to take them into account.

Conclusion

The theoretical reasons mentioned above were verified by a numerical experiment. As a result the 85,3 % decrease of the standard deviation $\sigma_{32}(r)$ was obtained, which is equivalent to the increase of the test sample from thirty-two experiments to one hundred and nine experiments. Hence, the development of a quantum corrector for the computational errors of correlation

coefficients is ca. eight times better than attempting to correct the data in the continuous (analog) form (4).

Actually we have obtained some continuous-quantum oracle which predicts a correlation coefficient better than the Pearson formula (3). This quantum oracle implements much more complicated computations according to the program which consists of approximately 250 lines of mathematical high-level language code. Data processing by this program has a linear computational complexity and takes about two seconds of machine time. Actually the quantum oracle is created which predicts the value of the correlation coefficient for the samples of thirty-two experiments. The oracle allows one to exchange the computational complexity for the precision of the estimation.

The increase of the computational complexity from linear to quadratic is assumed to lead to a significant increase of precision for prediction of the correlation coefficient in the small samples case. It is extremely important for development of the series of algorithms like GOST R 52633.5 for the neural net biometry-code convertor automatic training.

References

1. GOST R 52633.5–2011 Information protection. Information protection technology. The neural net biometry-code convertor automatic training.

2. Serikova, Yu. I. Compensation of methodological errors in calculations of standard deviations and correlation coefficients occuring due to small sample sizes // University proceedings. Volga region. Engineering sciences / Yu. I. Serikova, V. I. Volchikhin, A. I. Ivanov. – 2016. – № 1. – P. 45–49.

3. Serikova, Yu. I. Compensation of methodological and random components of errors in calculations of correlation coefficients occuring due to small sample sizes / Yu. I. Serikova, V. P. Kulagin, A. I. Ivanov // Information Technology. – 2016. – Vol. 22, № 9. – P. 705–710.

4. Volchikhin, V. I. Fractal-correlation functional used in the search for pairs of weakly dependent biometric data in small samples / V. I. Volchikhin, A. I. Ivanov, B. B. Akhmetov, Yu. I. Serikova // University proceedings. Volga region. Engineering sciences. – 2016. – № 4. – P. 25–31.

5. Nielsen, M. A. Quantum computation and quantum information / M. A. Nielsen, I. L. Chuang. – Cambridge University Press, 2000. – P. 1–710.

6. Serikova, Yu. I. Quantum superposition of the discrete spectrum of mathematical correlation molecule stat us for small samples of biometric data / Yu. I. Serikova, V. I. Volchikhin, A. I. Ivanov, A. V. Serikov // Mordovia University Bulletin. – 2017. – Vol. 27, № 2. – P. 224–238.

И. Ю. Шполянская, А. Р. Загиров

Ростовский государственный экономический университет, Ростов-на-Дону, Россия

РАЗРАБОТКА WEB-ОРИЕНТИРОВАННОЙ РЕКОМЕНДАТЕЛЬНОЙ СИСТЕМЫ С ПРИМЕНЕНИЕМ ТЕХНОЛОГИЙ WEB MINING

В условиях растущей конкуренции предприятия все чаще используют web-ресурсы для создания CRM систем для управления взаимоотношениями с клиентами в режиме онлайн. В докладе представлена модель web-ориентированной рекомендательной системы, разрабатываемой на основе технологий Web Mining. Система позволяет построить систему рекомендаций клиентам на основе оперативного учета и анализа их поведения на сайте.

Одной из главных функций управления предприятием является управление взаимоотношениями с клиентами. В этих целях целесообразно создание CRM-системы, позволя-

ющей компании эффективно управлять своими взаимоотношениями с клиентами для успешного развития в условиях конкуренции. Такая система направлена на удовлетворение запросов клиента, на привлечение новых и удержание существующих клиентов предприятия. Актуальной является задача создания CRM-систем в виде независимых интеллектуальных Web-сервисов, которые при минимальной стоимости могли обеспечить реализацию основных целей стратегии управления взаимодействием с клиентами на основе современных технологий интеллектуального анализа Web Mining [1].

Цель внедрения web-ориентированной CRM-системы – оптимизация работы с клиентами на основе ресурсов сайта компании, в том числе: продвижение товаров и услуг в сети; определение целевой аудитории; анализ структуры запросов клиентов увеличение удовлетворенности и лояльности клиентов; и др. Основными функциями web-ориентированной CRM-системы для эффективного управления взаимодействием с клиентами становятся управление потенциальными сделками, том числе ведение информационной базы о продуктах и услугах компании, создание и управление списком потенциальных клиентов, сегментация клиентской базы, формирование персонализированных предложений.

Целью создания web-CRM является формирование клиентской базы на основе сбора и анализа данных в режиме онлайн для выявления информации о наиболее ценных клиентах, а также разработка стратегии, какими методами компания должна взаимодействовать с ними. В web-CRM данные о клиенте, собранные на сайте, используется для более точного формирования предложения конкретному клиенту. Для этого в режиме онлайн формируется информационная база, содержащая оперативные данные о поведении и предпочтениях пользователей в процессе взаимодействия с web-ресурсом, а также обо всех сделках с клиентами с их характеристиками. На основе актуализированной базы о клиентах, их сделках и предпочтениях проводится анализ данных. Основными задачами анализа данных в CRM являются следующие: анализ запросов и предпочтений клиентов; сегментация клиентов; выделение целевой группы клиентов; выбор способов взаимодействия с клиентом и формирование рекомендаций целевой группе на основе сформированных кластеров.

В этих целях в состав CRM-систем могут быть включены процедуры интеллектуального анализа данных – Data Mining. В основе инструментов Data Mining лежат технологии машинного обучения, в том числе: деревьев решений; ассоциативных правил; генетических алгоритмов; нейронных сетей и др. Процедуры Data Mining в составе аналитических CRM-систем основаны на использовании ряда математических и статистических методов классификации, кластеризации, прогнозирования, поиска зависимостей и ассоциативных правил [2].

Процедуры классификации в CRM позволяют решить задачи сегментации и оценки перспективности клиентов; выделение групп однородных с точки зрения интересов пользователя товаров и услуг для формирования эффективной стратегии взаимодействия с клиентом, прежде всего путем выработки соответствующих рекомендаций пользователю ресурса.

Процедуры регрессионного анализа позволяют решать задачи: прогнозирование спроса клиента из определенной группы на тот или иной вид продукции или услуг; оценка вероятности повторных заказов; анализ влияния различных факторов на спрос на продукцию и услуги и др.

Процедуры выявления ассоциаций (анализ событий, происходящих сов-местно), позволяют предсказывать поведение клиента, формировать предложение продукции и услуги, которые, вероятно, его заинтересует; реализовать кросс-продажи различных видов товаров и услуг.

Основные задачи управления взаимоотношениями с клиентам, которые целесообразно решать в web CRM-системах с помощью инструментов Data Mining [1]:

– идентификация клиента – отслеживание поведения и предпочтений, а также определение ценности клиента для предприятия (задача кластеризации);

– сегментирование клиентов. Выделение группы ценных клиентов, которые приносят наибольший доход компании (задача классификации);

– взаимодействие с клиентом – выработка индивидуальной стратегии обслуживания пользователя сайта на основе оценки его поведения, предпочтений и потребностей (ассоциативные правила).

Модель управления в системе Web CRM представлена в следующем виде:

1. Сбор на сайте и формирование данных, касающихся клиентов и их предпочтений относительно различных видов товаров и услуг. Структурирование этих данных, чтобы определить конкретные группы клиентов с однотипными запросами. Для этого используются методы Web Mining, чтобы извлечь из имеющихся больших массивов данных полезную информацию и в соответствии с ней классифицировать клиентов по группам.

2. На основе сформированных групп определяются ценные потребительские сегменты. Цель этапа состоит в том, чтобы определить, персонализировать отношение к каждой группе в соответствии с конкретными потребностями клиентов и их значимостью для компании.

3. Формирование эффективной стратегии взаимодействия с клиентом на основе информации, полученной на предыдущих этапах [3].

В отличие от Data Mining, где извлечение данных работает в режиме offline, в Web Mining анализ происходит в режиме online. В Data Mining данные хранятся в базах данных, а в Web Mining – в базе данных сервера и веб-журнале.

В бизнес-аналитике Web Mining позволяет решить задачи, которые имеют важное значение определения онлайн поведения пользователей – клиентов: последовательность просматриваемых пользователями страниц сайта; анализ поведения, запросов и предпочтений клиентов; поиск зависимостей при использовании пользователем ресурсов сайта (поиск ассоциативных правил); сбор веб-статистики; оптимизация структуры веб-ресурса; оптимизация размещения описания товаров и услуг на страницах сайта; персонализация контента страницы сайта; персонализация предоставления продуктов, услуг.

Одним из основных применений Web Mining является формирование полезной информации об интернет-клиентах: их характеристики, потребности, интересы и поведение. В контексте Web CRM, целью которых является определение конкретных групп клиентов в соответствии с такими критериями, как поведение, это может служить в качестве основы для выработки рекомендаций для определенных целевых групп клиентов.

Другая возможность заключается в использовании полученных результатов анализа непосредственно для оптимизации интернет-портала компании. После сбора информации о клиентах и определения предпочтительного способа просмотра страниц сайта можно оптимальным образом разместить информацию о продукте на соответствующих важных страницах. Кроме того, содержание веб-страницы может отображаться динамически для каждого клиента в зависимости от группы, к которой он принадлежит.

На рис. 1 представлена концептуальная модель веб-сервиса CRM со встроенной системой рекомендаций.

В процессе работы системы для связывания групп пользователей с категориями товаров или услуг используются технологии Semantic Web. Данные о пользователях и их группах, товары и услуги, а также рекомендованные категории товаров определенным группам пользователей представляются в виде RDF-графов. Это позволяет легко модифицировать структуру и описание групп в соответствии с реакциями пользователей на сайте, а также находить точное соответствие каждой группе категорию рекомендуемых продуктов [4]. На рис. 2 приведен фрагмент онтологии веб-сервиса с группами товаров интернет-магазина.

Созданная онтологическая модель дает целостное описание структуры информационных ресурсов системы и позволяет осуществить максимально возможное сохранение семантических связей при обработке информации о клиенте. Использование данного подхода позволяет создать систему выработки рекомендаций, способную к адаптации и изменению в процессе ее работы.

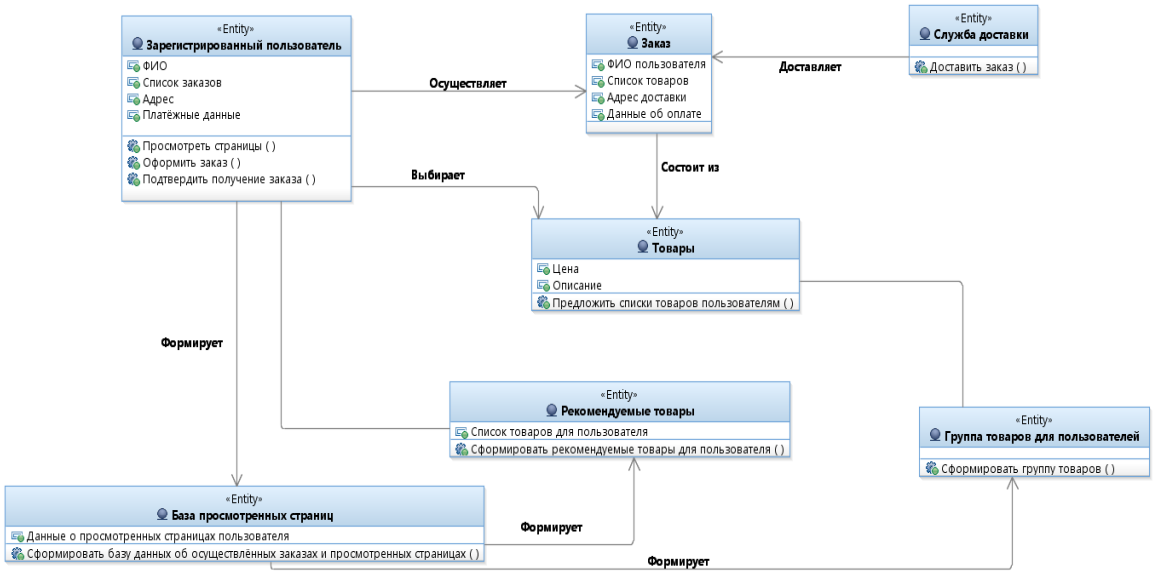


Рис. 1. Концептуальная модель веб-сервиса CRM

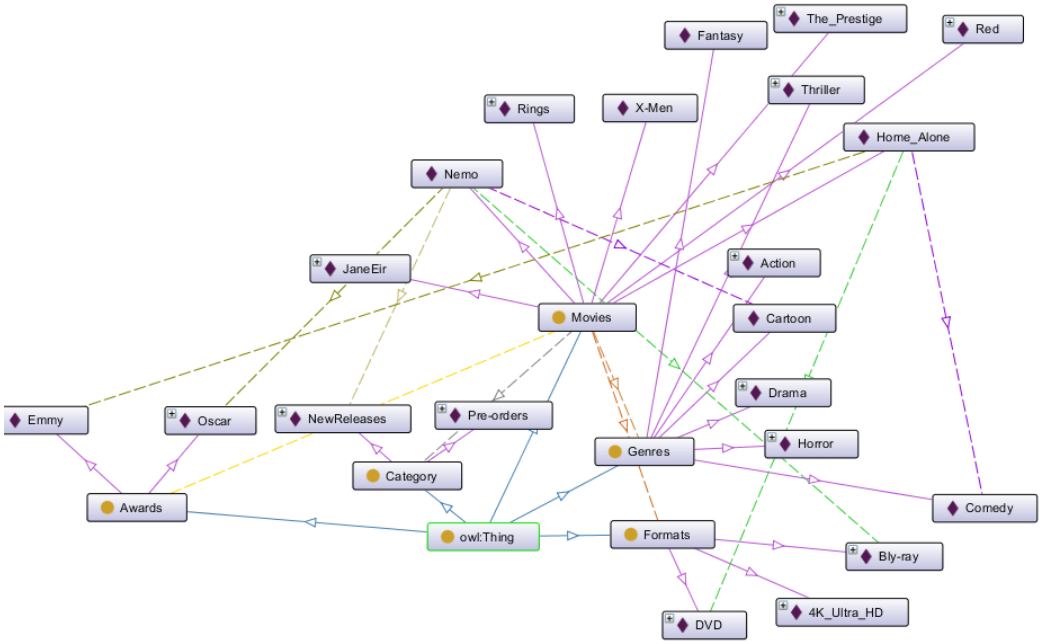


Рис. 2. Фрагмент онтологии веб-сервиса CRM с группами товаров интернет-магазина

Библиографический список

1. Шполянская, И. Ю. Использование технологий Data mining для создания аналитических CRM-систем для малого бизнеса / И. Ю. Шполянская // Вестник Ростовского государственного экономического университета (РИНХ). – 2014. – № 1 (45). – С. 129–135.

2. Киселев, М. Средства добычи знаний в бизнесе и финансах / М. Киселев, Е. Соломатин // Открытые системы. – 1997. – № 4. – С. 41–44.

3. Liu, B. Web Usage Mining. Web Data Mining. Exploring Hyperlinks, Contents, and Usage Data / B. Liu, B. Mobasher, O. Nasraoui ; ed. by Liu B. – Springer-Verlag, 2011. – P. 527–603.

4. Шполянская, И. Ю. Референтная онтологическая модель бизнеса как основа создания WEB-ориентированных систем и сервисов / И. Ю. Шполянская // Вестник Ростовского государственного экономического университета (РИНХ). – 2015. – № 2 (50). – С. 220–226.

П. П. Макарычев

Пензенский государственный университет, Пенза, Россия

ЭВРИСТИЧЕСКИЕ АЛГОРИТМЫ РЕШЕНИЯ ОПТИМИЗАЦИОННЫХ ЗАДАЧ В ОБЛАСТИ ИНФОРМАТИКИ

Обсуждается подход к разработке эвристических алгоритмов на основе тензорной (матричной) методологии.

Эвристический алгоритм не гарантирует точного (оптимального) решения, но является достаточно эффективным для задач высокой вычислительной сложности. Основные отличия эвристического алгоритма от теоретически обоснованного: не гарантирует нахождение наилучшего решения; не гарантирует нахождения решения, если оно заведомо существует; можно найти неверное решение [1]. В данной работе рассматриваются эвристические алгоритмы, разработанные автором для решения задач коммивояжера и назначений [2, 3].

Алгоритм решения задачи коммивояжера. Предположим, что дано множество маршрутов коммивояжера S и числовая целевая функция $U(s^*)$. Требуется найти маршрут $s^* \in S$, значение $U(s^*)$ для которого будет наименьшим (оптимальным) для всех $s \in S$.

Пусть матрица расстояний между городами v_1, v_2, v_3, v_4, v_5 имеет вид:

$$\mathrm{M}_0 = \begin{bmatrix} \infty & 9 & 3 & 5 & 6 \\ 9 & \infty & 2 & 9 & 4 \\ 3 & 2 & \infty & 7 & 7 \\ 5 & 9 & 7 & \infty & 3 \\ 6 & 4 & 7 & 3 & \infty \end{bmatrix}.$$

Строки и столбцы матрицы M_0 соответствуют городам. Преобразуем матрицу M_0 к виду, при котором расстояния между городами начального маршрута расположены на главной диагонали:

$$\mathrm{M}_1 = \begin{bmatrix} \mathbf{6} & 4 & 7 & 3 & \infty \\ \infty & \mathbf{9} & 3 & 5 & 6 \\ 9 & \infty & \mathbf{2} & 9 & 4 \\ 3 & 2 & \infty & \mathbf{7} & 7 \\ 5 & 9 & 7 & \infty & \mathbf{3} \end{bmatrix}. \qquad (1)$$

Согласно (1) коммивояжер выезжает из города v_5 в город v_1. Последовательно объезжает города v_1, v_2, v_3, v_4 и возвращается в город v_5. При этом значение целевой функции $U(s)$ определяется следом тензора (матрицы) $tr(\mathrm{M}_1)=27$. Для решения задачи поиска минимума целевой функции будем искать последовательность матриц $\mathrm{M}_2, \mathrm{M}_3, \ldots$, для которых выполняется следующее условие:

$$\mathrm{tr}(\mathrm{M}_1) > tr(\mathrm{M}_2) > tr(\mathrm{M}_3) > \ldots .$$

Например, в матрице M_1, можно выделить фрагмент маршрута

$$\{(v_5, v_1), (\mathbf{v_1}, \mathbf{v_2}), (\mathbf{v_2}, \mathbf{v_3}), (\mathbf{v_3}, \mathbf{v_4}), (v_4, v_5)\} \in S,$$

длина которого $L((v_1,v_2),(v_2,v_3),(v_3,v_4))=18$. Произведем перестановку элементов в выделенном фрагменте маршрута:

$$\{(v_5,v_1),(\mathbf{v_1},\mathbf{v_3}),(\mathbf{v_3},\mathbf{v_2}),(\mathbf{v_2},\mathbf{v_4}),(v_4,v_5)\}\in S. \quad (2)$$

Из выражений (2) следует, что центральный элемент последовательности, в соответствие с выражением (1), не преобразуется $(\mathbf{v_2},\mathbf{v_3})=(\mathbf{v_3},\mathbf{v_2})$. Этот элемент вносит один и тот же вклад в длину маршрута коммивояжера. Поскольку длина фрагмента маршрута при этом составляет величину равную 14, что значительно меньше длины исходного фрагмента маршрута, то маршрут коммивояжера может быть изменен и представлен матрицей, имеющей след $tr(\mathrm{M}_2)=23$:

$$\mathrm{M}_2=\begin{bmatrix}\mathbf{6} & 7 & 4 & 3 & \infty\\ \infty & \mathbf{3} & 9 & 5 & 6\\ 3 & \infty & \mathbf{2} & 7 & 7\\ 9 & 2 & \infty & \mathbf{9} & 4\\ 5 & 7 & 9 & \infty & \mathbf{3}\end{bmatrix}.$$

В матрице M_2, характеризующей новый маршрут коммивояжера, можно осуществит поиск фрагмента маршрута с двумя центральными элементами $(\mathbf{v_1},\mathbf{v_3}),(\mathbf{v_3},\mathbf{v_2})$:

$$\{(\mathbf{v_5},\mathbf{v_1}),(\mathbf{v_1},\mathbf{v_3}),(\mathbf{v_3},\mathbf{v_2}),(\mathbf{v_2},\mathbf{v_4}),(v_4,v_5)\}\in S$$

Длина фрагмента маршрута составляет величину

$$L((v_5,v_1),(v_1,v_3),(v_3,v_2),(v_2,v_4))=20.$$

Произведем изменения в выделенном фрагменте маршрута:

$$\{(\mathbf{v_5},\mathbf{v_2}),(\mathbf{v_2},\mathbf{v_3}),(\mathbf{v_3},\mathbf{v_1}),(\mathbf{v_1},\mathbf{v_4}),(v_4,v_5)\}\in S \quad (3)$$

Из выражения (3) следует, что центральные элементы последовательности $(\mathbf{v_1},\mathbf{v_3}),(\mathbf{v_3},\mathbf{v_2})=(\mathbf{v_2},\mathbf{v_3}),(\mathbf{v_3},\mathbf{v_1})$ не преобразуются и вносят один и тот же вклад в длину маршрута коммивояжера. Длина фрагмента маршрута после преобразования равна $tr(\mathrm{M}_3)=14$. Следовательно, маршрут коммивояжера может быть изменен и представлен матрицей:

$$\mathrm{M}_3=\begin{bmatrix}\mathbf{4} & 7 & 6 & 3 & \infty\\ \infty & \mathbf{2} & 9 & 9 & 4\\ 2 & \infty & \mathbf{3} & 7 & 7\\ 9 & 3 & \infty & \mathbf{5} & 6\\ 9 & 7 & 5 & \infty & \mathbf{3}\end{bmatrix}. \quad (4)$$

Матрица M_3 имеет минимальное значение следа $tr(\mathrm{M}_3)=17$. Следовательно, маршрут коммивояжера (4) является оптимальным.

Алгоритм решения задачи назначения. Управление средствами поражения подразумевает решение задачи о назначении целей. К настоящему времени предложены решения задачи на основе различных подходов: конструктивный алгоритм, алгоритм исчерпывающего поиска, венгерский метод, комбинаторная оптимизация и т.д. [3].

Предположим, что имеется множество средств поражения – $W=\{w_1,w_2,...,w_n\}$, множество целей – $C=\{c_1,c_2,...,c_m\}$. Каждое средство поражения имеет g целевых кана-

лов и может быть назначено на любую цель. Каждая цель характеризуется уровнем угрозы $U_i, i = 1,2,...,m$ и вектором вероятностей (степеней) поражения [3]:

$$p_i = (p_{i,1},..., p_{i,j},..., p_{i,n}).$$

На основе вероятностей поражения для целей формируется вектор:

$$q_i = (q_{i,1}, q_{i,2},..., q_{i,n}) = ((1-p_{i,1})^{x_{1j}}, (1-p_{i,2})^{x_{ij}},..., (1-p_{i,n})^{x_{mj}}),$$

где $x_{i,j}$ – количество назначенных целевых каналов по каждой цели.

Другими словами, при появлении новой цели для всех средств поражения, производится оценка остаточной угрозы в виде:

$$U(c_i) = (U_i q_{i,1}, U_i q_{i,2},..., U_i q_{i,n}).$$

Пусть $m = 9, n = 3, g = 3$ тензор остаточных угроз $\mathbf{A}$, тензор назначений $\mathbf{B}$ в матричной форме представления имеют вид:

$$\mathbf{A} = \begin{bmatrix} \mathbf{1} & 2{,}1 & 3 \\ 2{,}5 & 2 & \mathbf{1{,}2} \\ 4 & \mathbf{2} & 3 \\ \mathbf{1{,}4} & 2 & 2 \\ \mathbf{3} & 3{,}4 & 4 \\ 4{,}1 & 5{,}6 & \mathbf{4{,}7} \\ 2{,}9 & 5 & \mathbf{3}{,}2 \\ 5{,}3 & \mathbf{3} & 2{,}2 \\ 2{,}1 & \mathbf{5} & 3{,}4 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix}.$$

В тензоре $\mathbf{A}$ элементы, отражающие начальное назначение целей средствам поражения отмечены жирным шрифтом. Несложно убедиться, что назначение целей осуществлено с использованием конструктивного (жадного) алгоритма. В целом результаты назначения можно представить тензором $\mathbf{B}$. В тензоре $\mathbf{B}$ столбцы соответствуют целям, строки – средствам поражения целей. Из $\mathbf{A}$ следует, что, с увеличением числа целей $(5 \le i \le 9)$, конструктивный алгоритм назначения не отвечает требованию минимума остаточной угрозы, создаваемой всей совокупностью целей.

На основе тензоров $\mathbf{A}$, $\mathbf{B}$ может быть определен тензор

$$\mathbf{C} = \mathbf{AB} = \begin{bmatrix} \mathbf{1} & 3 & 2{,}1 & 1 & 1 & 3 & 3 & 2{,}1 & 2{,}1 \\ 2 & \mathbf{1{,}2} & 2 & 2 & 2 & 1{,}2 & 1{,}2 & 2 & 2 \\ 4 & 3 & \mathbf{2} & 4 & 4 & 3 & 3 & 2 & 2 \\ 1{,}4 & 2 & 2 & \mathbf{1{,}4} & 1{,}4 & 2 & 2 & 2 & 2 \\ 3 & 4 & 3{,}4 & 3 & \mathbf{3} & 4 & 3 & 3{,}4 & 3{,}4 \\ 4 & 4{,}7 & 3{,}6 & 4 & 4 & \mathbf{4{,}7} & 4{,}7 & 3{,}6 & 3{,}6 \\ 3 & 3{,}2 & 5 & 3 & 3 & 3{,}2 & \mathbf{3{,}2} & 5 & 5 \\ 5 & 2{,}2 & 3{,}5 & 5 & 5 & 2{,}2 & 2{,}2 & \mathbf{3}{,}5 & 3{,}5 \\ 2 & 3{,}4 & 5 & 2 & 2 & 3{,}4 & 3{,}4 & 5 & \mathbf{5} \end{bmatrix}. \qquad (5)$$

В тензоре $\mathbf{C}$ строки соответствуют целям, столбцы средствам огневого поражения и целевым каналам. На главной диагонали выделены значения остаточных угроз сформированных назначениях. Для конкретизации назначений на основе тензора $\mathbf{B}$ можно сформировать вектор

$$\mathbf{V} = (1 \quad 3 \quad 2 \quad 1 \quad 1 \quad 3 \quad 3 \quad 2 \quad 2).$$

Значение элемента вектора $\mathbf{V}$ соответствует номеру средства поражения, положение элемента среди элементов с равными значениями – номеру целевого канала. След тензора характеризует суммарную остаточную угрозу от целей $tr(\mathbf{C}) = 24{,}3$ (функция цели) при выполненном назначении.

Абсолютное значение суммарной остаточной угрозы не характеризует качество выполненного назначения целей. Поэтому следует исследовать окрестности в пространстве состояний, определить и устранить все несоответствия в назначении целей с использованием формулы:

$$f_{i,j} = \begin{cases} 1, \text{если } T_{i,i} + T_{j,j} > T_{i,j} + T_{j,i} \\ 0, \text{иначе.} \end{cases}, i = 1,...,m;\ j = 1,...,m \qquad (6)$$

Нулевое значение $f_{i,j}$ является индикатором выявленного несоответствия в назначении двух целей.

Из (5, 6) следует, что имеет место несоответствия между 1 и 9 целями, 6 и 8 целями и другие несоответствия. Для устранения названных первых двух несоответствий можно задать тензор $\mathbf{G}$ и выполнить следующее преобразование тензора $\mathbf{C}$

$$\mathbf{C} = \mathbf{C} \cdot \mathbf{G} = \begin{bmatrix} \mathbf{2{,}1} & 3 & 2{,}1 & 1 & 1 & 2{,}1 & 3 & 3 & 1 \\ 2 & \mathbf{1{,}2} & 2 & 2 & 2 & 2 & 2 & 1{,}2 & 2 \\ 2 & 3 & \mathbf{2} & 4 & 4 & 2 & 2 & 3 & 4 \\ 2 & 2 & 2 & \mathbf{1{,}4} & 1{,}4 & 2 & 2 & 2 & 1{,}4 \\ 3{,}4 & 4 & 3{,}4 & 3 & \mathbf{3} & 3{,}4 & 3 & 4 & 3 \\ 4{,}7 & 4{,}7 & 5{,}6 & 4 & 4 & \mathbf{4{,}7} & 5 & 4{,}7 & 4 \\ 5 & 3{,}2 & 5 & 3 & 3 & 5 & \mathbf{3{,}2} & 3{,}2 & 3 \\ 3 & 2{,}2 & 3 & 5 & 5 & 3 & 2{,}2 & \mathbf{2{,}2} & 5 \\ 5 & 3{,}4 & 5 & 2 & 2 & 5 & 3{,}4 & 3{,}4 & \mathbf{2} \end{bmatrix}$$

После выполнения преобразования значение следа тензора $tr(\mathbf{C}) = 21{,}8$. Таким образом, осуществляя последовательно локальный поиск и устраняя все несоответствия в назначении, можно найти минимальное значение следа тензора. На основе рассмотренных алгоритмов разработаны программы, зарегистрированные в Федеральной службе по интеллектуальной собственности.

Библиографический список

1. Ватутин, Э. И. Основы дискретной комбинаторной оптимизации / Э. И. Ватутин, В. С. Титов, С. Г. Емельянов. – М. : Аргамак-Медиа, 2016. – 270 с.

2. Макарычев, П. П. Комбинаторная оптимизация пути коммивояжера с применением тензорной методологии / П. П. Макарычев // Аналитические и численные методы моделирования естественнонаучных и социальных проблем : сб. материалов XI Междунар. науч.-техн. конф. – Пенза, 2016. – С. 85–88.

3. Макарычев, П. П. Решение задачи о назначении целей на основе тензорной методологии / П. П. Макарычев, В. С. Безяев // Новые информационные технологии и системы : сб. науч. ст. XIII Междунар. науч.-техн. конф. – Пенза, 2016. – С. 27–29.

СИСТЕМЫ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ

В. В. Эпп, И. П. Бурукина

Пензенский государственный университет, Пенза, Россия

АЛГОРИТМЫ РАЗМЕЩЕНИЯ И ТРАССИРОВКИ В МОБИЛЬНЫХ ИГРАХ

Рассмотрены вопросы автоматизации размещения элементов игровых уровней в мобильных играх с использованием известных алгоритмов автоматизированного проектирования печатных плат.

Существует множество различных жанров игр, которым необходимо постоянно генерировать новые уровни игры. Такие игры не могут ограничиваться конечным набором уровней, так как несут в себе соревновательный характер и содержат таблицу рекордов. При ограниченном наборе уровней таблица достаточно быстро заполнится максимальными результатами, и интерес к приложению пропадет. Процесс создания уровней требует немало времени сил и дополнительных затрат, а в результате выходит лишь ограниченный набор уровней.

Главная проблема заключается в том, что уровни занимают место, а на мобильных устройствах оно ограничено. Генерация легко решает эту проблему, так как отпадает надобность хранить уровни на диске. Уровни создаются во время игрового процесса и затем, при его завершении, стираются из памяти. Сама идея генерации уровней так же подразумевает и удобное средство для разработчиков, позволяющее автоматизировать процесс создания уровней для игр, относящихся к выбранному жанру, так как проще адаптировать код генератора для своих целей, чем создавать десятки уровней вручную.

За всю историю игровой индустрии было разработано немало алгоритмов генерации уровней. Хотелось бы отметить такие как: алгоритм, основанный на BSP – деревьях, алгоритм Random Room Placement, алгоритм Procedurally Build. Рассмотрим особенности этих алгоритмов более подробно [2].

Самым простым алгоритмом является алгоритм Random Room Placement. Данный алгоритм работает с двумерным массивом. Для начала в случайных местах расставляются комнаты разных размеров до тех пор, пока не будет достигнуто заданное число комнат. Если две комнаты пересекаются, то новая комната удаляется, и процесс генерации повторяется до тех пор, пока комната не будет установлена. Затем комнаты соединяются произвольно коридорами. Алгоритм пригоден для генерации заготовок уровней, но требует доработки для расставления игровых элементов, а так же не подходит для мобильных платформ ввиду больших временных затрат.

Алгоритм Procedurally Build – на первом этапе массив заполняется стенами, затем происходит заполнение области комнатами по схожей с предыдущим алгоритмом схеме, но при этом расстановка происходит только в тех областях, где находятся стены, таким образом быстродействие получается выше. Объединение комнат осуществляется следующим образом: каждая комната при помощи алгоритма поиска кратчайшего пути (как правило алгоритм A*) соединяется с другой комнатой.

Алгоритм с использованием BSP-деревьев. Принцип генерации при помощи данного алгоритма отличается от двух предыдущих. Генерация осуществляется в некоторой прямоугольной области, как правило, экранной. Область разделяется на две части. Затем случайная из двух частей делится так же на части, и так несколько раз. Далее в каждой области

генерируется прямоугольная комната, не выходящая за границы своей области, одновременно строится дерево комнат. После генерации областей происходит построение всех возможных путей между ними. На последнем этапе тупиковые и циклические пути удаляются. Основным недостатком этого алгоритма является время генерации.

В процессе рассмотрения существующих алгоритмов были выделены следующие основные недостатки:

- вероятность зацикливания;
- низкая скорость генерации;
- не расставляет игровые элементы.

Для устранения вышеизложенных недостатков было предложено использовать хорошо зарекомендовавшие себя алгоритмы размещения и трассировки, которые используются при проектировании печатных плат, а именно волновой алгоритм Ли и алгоритм размещения разно габаритных элементов [1].

Принцип работы получившегося алгоритма генерации уровней RPG игр заключается в следующем:

- помещаем в open-список стартовый узел;
- основной цикл (пока open список не пуст);
- берем из open списка верхний узел;
- если этот узел равен конечному, то строим путь и выходим из основного цикла;
- обрабатываем связи выбранного узла. Для каждого соседа, а именно проверяем его на проходимость, для конкретного юнита, если непроходим, то, понятное дело, переходим к следующему соседу и вычисляем оценку пути от стартового узла через текущий до соседа;
- ищем соседа в open и close списках, если найден, то сравниваем его оценку пути от стартового узла с вычисленной, если его оценка лучше то идем к следующему соседу, в противном случае удаляем узел из соответствующего списка, пересчитываем оценку пути до конечного узла и помещаем этого соседа в open список, в соответствующее место;
- помещаем текущий узел в close список;
- переходим к п. 3 (конец основного цикла).

Если путь найден, его можно обработать, сделать более плавным, если это нужно. Самые тяжелые по быстродействию места в алгоритме – это поиск узла в списках. Если использовать стандартные списки из stl, то затраты составляют примерно 7 %, от времени работы всего алгоритма, это верно для достаточно сложной карты и поиске пути на значительное расстояние. Вставка узла в open список примерно 15 % и проверка проходимости (учитывая размеры юнита) – порядка 10 %.

В процессе разработки и тестирования прототипа работы алгоритма были обнаружены так называемые «особые случаи», которые необходимо выявлять и обрабатывать. Под «особыми случаями» понимаются следующие ситуации:

- нахождение комнаты вне зоны досягаемости остальных комнат;
- нахождение группы комнат вне зоны досягаемости остальных комнат;
- пересечение коридоров.

Поиск пересекающихся коридоров – Лишние стены отсекаются, таким образом пересечение становится проходимым.

Поиск «одиноких» комнат – осуществляется проверка обработки комнат, не имеющих возможных соседей. Такая комната удаляется.

Проверка поиска пути осуществляется на сгенерированных уровнях. Начальная комната помечена иконкой игрока, конечная – иконкой выхода. На рис. изображены результаты работы модуля поиска пути. Путь выделяется зелеными квадратами.

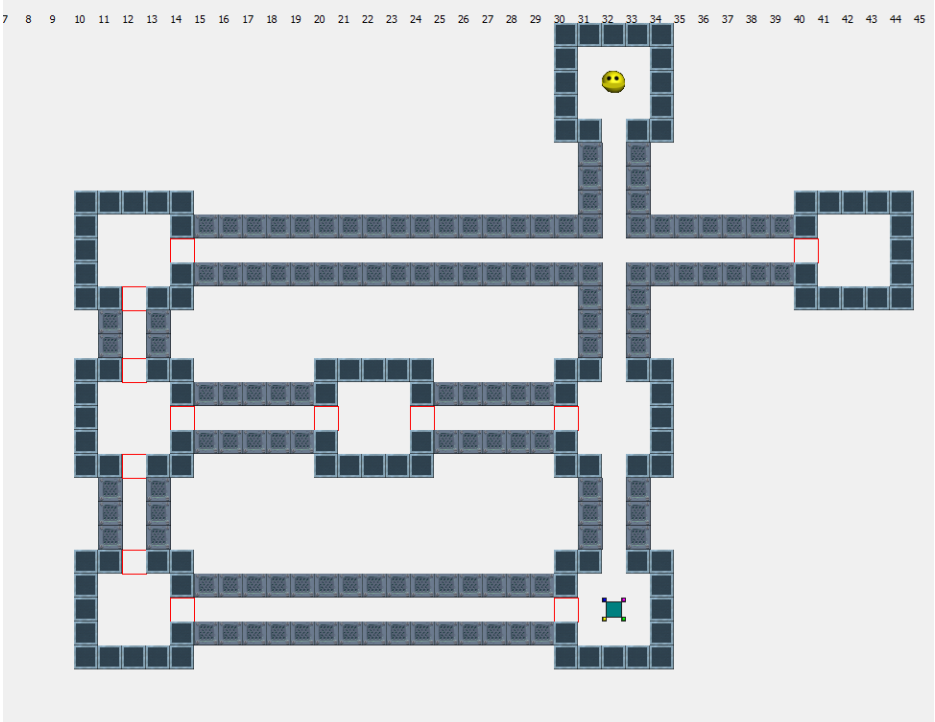


Рис. 1. Генерация пересечений

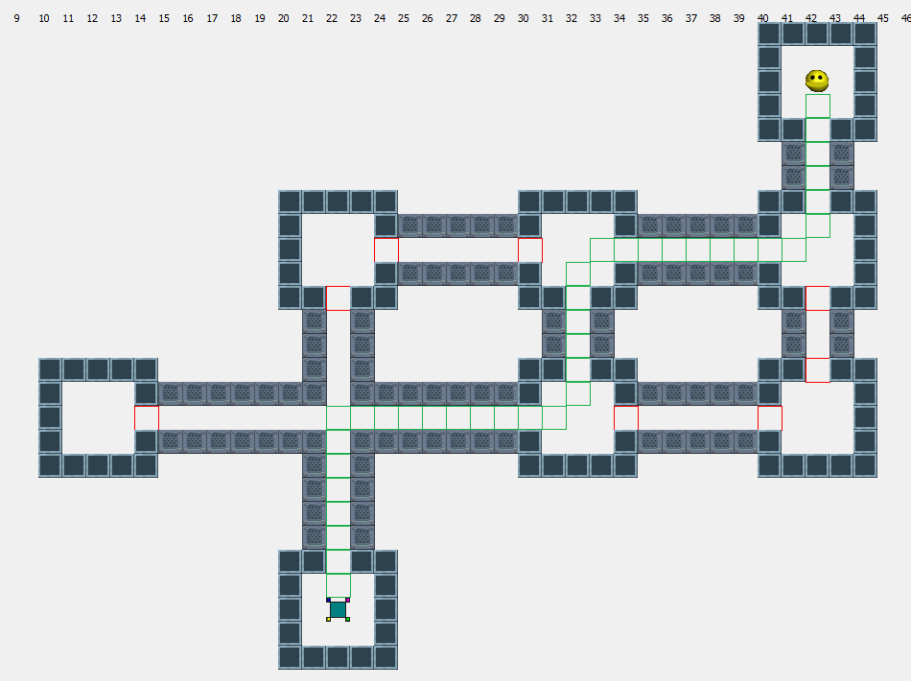


Рис. 2. Проверка алгоритма поиска пути

Библиографический список

1. Малика, А. С. Автоматизация конструирования РЭА / А. С. Малика, В. А. Деньдобренко. – М. : Высш. шк., 1988. – 304 с.

2. Орлов, С. Программная инженерия. Технологии разработки программного обеспечения / С. Орлов. – СПб. : Питер, 2016. – 640 с.

А. С. Войнов[1,2], И. В. Сенокосов[1] , Чен-Вэй Янг[2]

[1] Пензенский государственный университет, Пенза, Россия
[2] Технический университет Люлео, Люлео, Швеция

ТРАНСЛЯТОР SCL-ОПИСАНИЯ СЕТЕЙ SMART GRID В СИСТЕМЫ ФУНКЦИОНАЛЬНЫХ БЛОКОВ

Представлен транслятор исходного SCL-описания интеллектуальной электрической сети Smart Grid в набор взаимосвязанных функциональных блоков, которые полностью повторяют исходную структуру сети Smart Grid. В результате преобразования получается функционально-блочная модель сети Smart Grid для последующего ее использования при моделировании критических ситуаций в сетях электроснабжения (перегрузка, веерное отключение и т.д.).

В настоящее время все большую популярность приобретают электрические сети нового поколения – интеллектуальные электрические сети Smart Grid [1]. Введение нового стандарта электрификации IEC 61850 было необходимо ввиду повсеместного внедрения альтернативных источников энергии. Этот стандарт определяет интерфейсы для взаимодействия внутри сети, а также определяет концепцию моделирования и методологию проектирования сети.

В стандарте IEC 61850 представлен язык описания для Smart Grid – *System Configuration Language* (SCL), который базируется на XML.

Для автоматизации сетей Smart Grid зачастую используется стандарт IEC 61499, ориентированный на построение распределенных систем управления нового поколения. Основным артефактом проектирования в IEC 61499 являются функциональные блоки (ФБ). В данной статье

для работы с ФБ используется инструментальная система nxtStudio [4].

Рассмотрим подробнее структуру файла SCL:

1. Блок описания первичного оборудования подстанции.

Данный блок описывает элементы сети, отвечающие за генерацию электроэнергии.

2. Блок описания вторичного оборудования.

Данный блок описывает взаимосвязи и типы соединений между отдельными элементами сети.

3. Блок описания интеллектуальных электронных устройств (ИЭУ).

Данный блок описывает все ИЭУ сети, которые необходимы для ее управления. Совокупность этих устройств позволяет управлять сетью практически без участия человека.

4. Блок описания используемых типов данных.

Данный блок представляет собой описание форматов данных, используемых в Smart Grid. Это необходимо для возможности объединения в дальнейшем нескольких сетей в одну.

5. Блок описание протоколов взаимодействия по сети.

Данный блок содержит информацию об интерфейсах взаимодействия между отдельными ИЭУ посредством использования различных протоколов (например: TCP/IP).

В данной работе транслятор представляет собой программу (рис. 1), на вход которой подается файл SCL, а на выходе создается папка с готовым проектом nxtStudio, который содержит в себе функциональные блоки и HMI. Программа позволяет значительно сэкономить время проектирования системы управления сетью, поскольку к сгенерированным ФБ можно добавить специальные блоки управления.

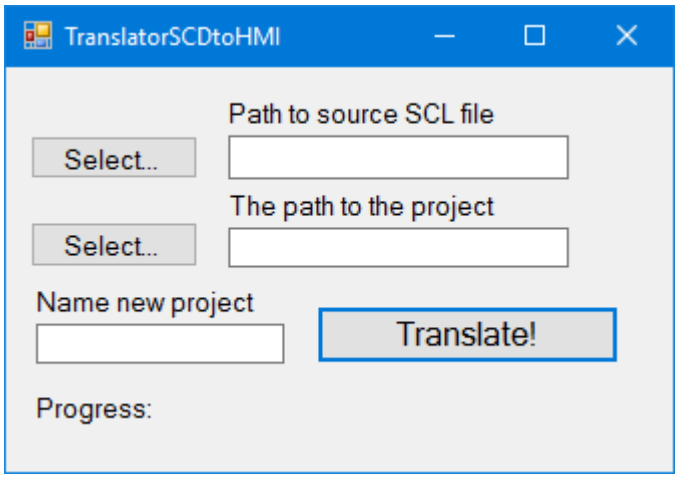


Рис. 1. Главное окно программы транслятора

Для начала трансляции необходимо ввести три основных параметра:

1. Путь до исходного SCL-файла.
2. Путь до пустой папки, в которую будет записан проект.

Шаблон для проекта nxtStudio содержится в самом трансляторе, что позволяет корректно выполнить трансляцию, поскольку встроенный в транслятор проект nxtStudio заранее подготовлен.

3. Имя нового проекта.

После того, как все исходные данные введены, пользователь должен нажать кнопку Translate и запустить работу транслятора. Во время работы в строке Progress будет отображаться текущая операция.

Рассмотрим основные этапы работы транслятора:

1. Создание нового проекта nxtStudio в папке, выбранной пользователем.

Для этого шаблон проекта просто копируется из памяти транслятора.

2. Расстановка всех элементов сети на плоскости.

На данном этапе определяются все устройства сети и располагаются на плоскости как часть человеко-машинного интерфейса (HMI).

3. Использование алгоритма Ли [5] для нахождения путей между всеми взаимосвязанными устройствами сети.

При этом важно отметить, что используется модифицированный алгоритм Ли. Основа модификации состоит в том, что у алгоритма появляется возможность разветвления для соединения трех и более элементов сети.

4. Создание функциональных блоков.

Для создания каждого функционального блока необходим следующий набор параметров:

– X,Y блока на HMI;

– начальные значения на входе в блок, если он представляет соеденения между элементами сети;

– полное имя устройства, на базе которого создан данный функциональный блок;

– уникальное имя данного функционального блока;

– уникальный идентификатор данного функционального блока, который необходим для его отображения на HMI;

– IP и номер порта для взаимодействия с устройством через сеть.

4. Запись сгенерированных функциональных блоков в проект nxtStudio.

Данный этап является завершающим при работе транслятора. До этого этапа никакие данные в проект nxtStudio не записываются. Это сделано для того, чтобы избежать ошибок в проекте в случае сбоя работы транслятора.

На рис. 2 представлен пример сети Smart Grid, описанной в SCL-файле и открытым в специальном редакторе VisualSCL[6].

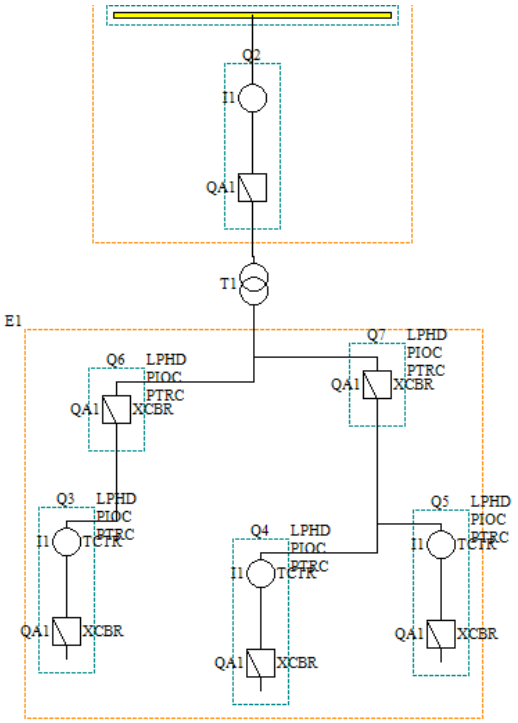


Рис. 2. Пример фрагмента исходной сети Smart Grid

После окончания работы транслятора будут сгенерированы все необходимые ФБ (рис. 3).

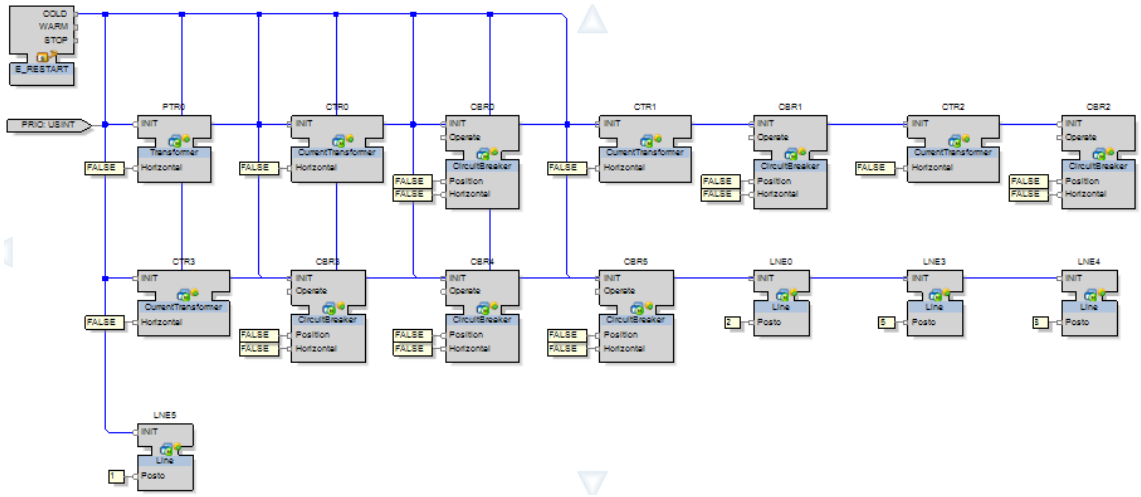


Рис. 3. Сеть функциональных блоков в среде nxtStudio, сгенерированная на основе фрагмента сети Smart Grid, представленного на рис. 2

Следует отметить, что транслятор был реализован программно на языке C/C# и был использован при разработке схемы управления сети Smart grid.

Библиографический список

1. Gharavi, H. Guest Editorial Special Section on Cyber-Physical Systems and Security for Smart Grid / H. Gharavi, H. Chen, C. Wietfeld // IEEE Transactions on Smart Grid. – 2015. – Vol. 6 (5). – P. 2405–2408.

2. IEC 61850 – Communication networks and systems for power utility automation, IEC Standard, Part 6: Configuration language for communication in electrical substations related to IEDs – Ed 2, 2009.

3. Christensen, J. H. IEC 61499 Architecture, Engineering Methodologies and Software Tools / J. H. Christensen // Knowledge and Technology Integration in Production and Services: Balancing Knowledge and Technology in Product and Service Life Cycle. – Boston, MA: Springer US, 2002. – P. 221–228.

4. nxtStudio. – URL: – http://www.nxtcontrol.com/

5. Lee, C. Y. An Algorithm for Path Connections and Its Applications / C. Y. Lee // IRE Transactions on Electronic Computers, 1961. – Vol. EC-10, № 3. – P. 346–365.

6. VisualSCL. – URL: http://ase-visual-scl.software.informer.com/1.0/

А. С. Войнов[1,2], Чен-Вэй Янг[2]

[1] Пензенский государственный университет, Пенза, Россия
[2] Технический университет Люлео, Люлео, Швеция

ИСПОЛЬЗОВАНИЕ АЛГОРИТМА ЛИ В ТРАССИРОВКЕ СОЕДИНЕНИЙ В ДИАГРАММАХ SLD

Рассматривается модификация алгоритма Ли (алгоритма поиска кратчайшего пути между двумя точками на плоскости) для возможности его использования при визуализации схем разумных сетей электроснабжения Smart Grid, построенных из исходных описаний на языке SLD.

Smart Grid [1] – интеллектуальная сеть электроснабжения нового поколения. Главное отличие от традиционной сети заключается в использовании интеллектуальных электронных устройств (ИЭУ), которые автоматически контролируют работу сети и исправляют ошибки, которые в ней возникают. Введение ИЭУ потребовалось ввиду главной особенности Smart Grid: использования альтернативных источников электроэнергии одновременно с традиционными. Альтернативные источники электроэнергии не имеют постоянных характеристик функционирования (при облачности электроэнергия, вырабатываемая солнечной батареей, будет значительно ниже, чем в ясный день). В масштабах огромной территории (области или страны) эту проблему можно решить, перераспределив электроэнергию в сети. Для этой цели и служат ИЭУ, которые автоматически это сделают без участия человека.

Концепция технологии Smart Grid описана в двух основных стандартах:

– IEC 61850 [2]. Этот стандарт определяет интерфейсы для взаимодействия внутри сети, а также определяет концепцию моделирования и методологию проектирования сети;

– IEC 61499 [3]. Этот стандарт определяет эталонную архитектуру распределенных систем управления на основе функциональных блоков, а также методы управления этой системой.

SDL (Specification and Description Language) – это язык моделирования объекта в виде набора диаграмм или схем. В стандарте IEC 61850 язык SDL представлен в виде System Configuration Language (SCL) в формате XML.

Алгоритм Ли[4] – это алгоритма поиска кратчайшего пути в лабиринте. В данной работе его модификация будет использоваться для нахождения кратчайшего пути между двумя объектами сети Smart Grid, представленными на плоскости.

Основные этапы работы традиционного алгоритма Ли:

1. Из исходной точки в двумерном пространстве по четырем направлениям начинает распространяться волна, увеличивая свою мощность на единицу каждую итерацию. Значение волны записывается при первом прохождении через данную точку и в дальней-

шем не изменяется, что позволяет представить область волны с одинаковым значением мощности в виде круга.

2. При достижении в ходе распространения волны конечной точки в пространстве, происходит ее остановка.

3. Последним этапом является определение пути между начальной и конечной точкой в пространстве. Для этого начинает распространяться обратная волна из конечной точки, но она проходит только по точкам, в которых имеется значение первой волны. Эта волна имеет начальную мощность, равную конечной мощности первой волны, и движется только в одном направлении – по убыванию значения мощности первой волны, пока не достигнет исходной точки.

Работа традиционного алгоритма Ли представлена на рис. 1.

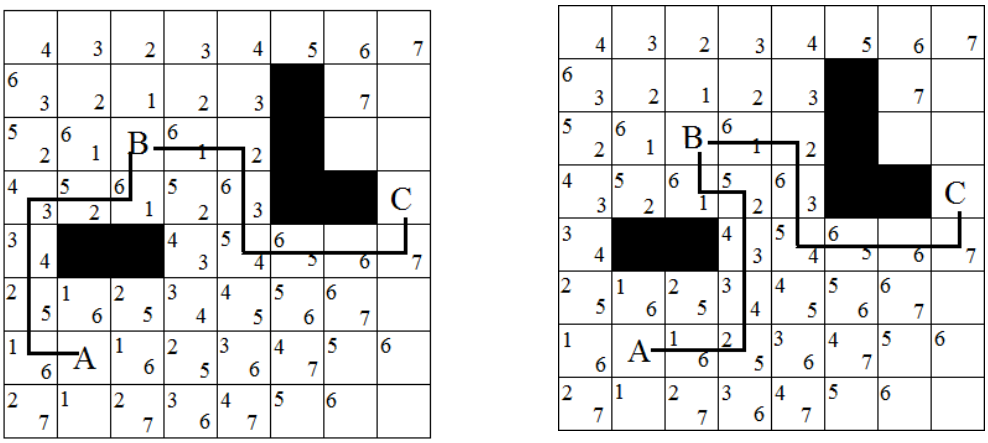


Рис. 1. Пример нахождения пути между точками A и B и точками B и C

В левом верхнем углу обозначены значения волны для пути между A и B, а в правом нижнем углу – между точками B и C. Черные квадраты обозначают препятствие. Рисунки справа и слева отображают различные варианты пути между точками A и B.

Как видно из рис. 1, существует несколько возможных кратчайших путей между точками A и B, из которых в традиционном алгоритме Ли путь выбирается случайным образом. Если рассмотреть пути на рис. 1 справа внимательнее, то можно заметить, что два различных пути можно заменить на один путь с разветвлением (рис. 2). Это особенно актуально при прокладке соединений между компонентами электрической сети, поскольку позволяет экономить на проводниках и обеспечивает уменьшение паразитных токов. Для этой цели была разработана модификация алгоритма Ли.

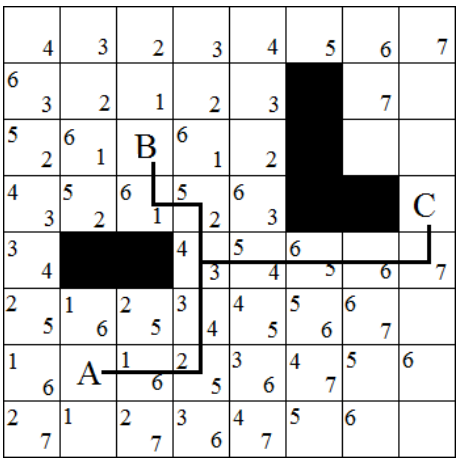


Рис. 2. Схема соединения между точками A, B, C используя модифицированный алгоритм Ли

Основные модификации алгоритма Ли:

1. Наличие приоритета по осям.

Движение второй волны всегда приоритетнее по оси *Y*. Это позволяет убрать путаницу, возникшую на рис. 1.

2. Приоритет по наименьшему числу связей.

Это обозначает, что начинать первую волну следует с точки, которая имеет меньше соединений. Например, необходимо соединить точку *A* с *B*, а точку *B* с *C*. Если путь между *A* и *B* уже существует, то соединение *B* с *C* следует делать начиная первую волну с точки *C*, потому что точка *B*, как и точка А, имеет уже одно соединение.

3. Наличие буфера памяти у каждой точки в пространстве, через которую может быть проложен путь.

Этот буфер используется только в том случае, если через данную точку проходит какой-либо путь. В этом случае в буфере храниться список точек, которые соединяет данный путь (две точки в простейшем случае). Каждая точка пути, показанного на рис. 2, содержит в своем буфере три точки (*A*, *B* и *C*). При работе модифицированного алгоритма Ли в дальнейшем будет проверятся не только достижение волной конечной точки, но и достижения точки, в буфере которой записана вершина, которою может достичь волна.

Рассмотрим пример применения модифицированного алгоритма Ли. Начальный файл описания сети Smart Grid (SLD-файл) был считан транслятором и все элементы, которые описываются в этом файле сети, были расположены на плоскости (рис. 3). Координаты элементов на плоскости вводятся также из SLD файла. Для удобства записи элементов в буфер ячейки при их соединении закодируем каждый из них уникальной буквой.

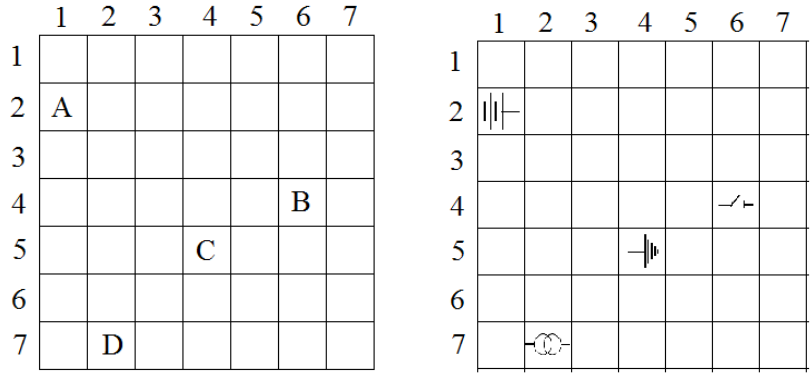


Рис. 3. Схема расположения элементов на плоскости и их кодировка перед началом работы модифицированного алгоритма Ли

Кроме того, из файла выбирается следующая информация о соединениях (для лучшего понимания приводится в формате координат с указанием стороны соединения (п-право, л- лево, в- верх, н- низ):

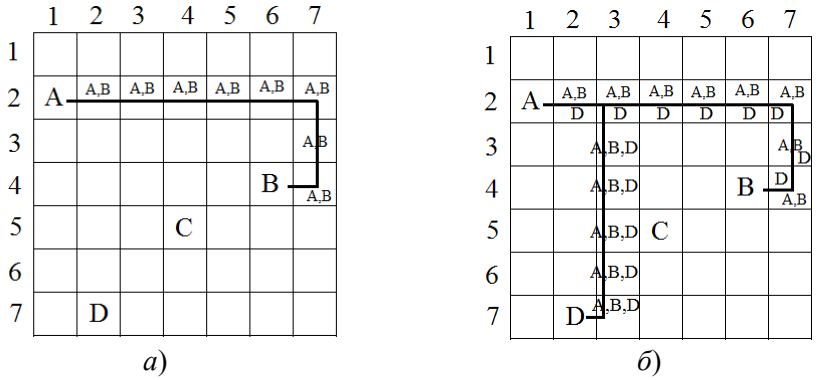
1) 1.2п – 6.4п ;
2) 1.2п – 2.7п;
3) 2.7л – 4.5л;
4) 4.5л – 6.5л.

Ход работы модифицированного алгоритма Ли для данного примера следующий:

1. Проводим первый путь между элементами сети и заносим в буфер каждой пройденной ячейки значение начального и конечного элемента сети (рис. 4,*а*).

2. Следующее соединение необходимо провести между точками *A* и *D*, но точка *A* уже имеет одно соединение, поэтому за начальную точку принимается точка *D*.

Распространение волны происходит до координаты (2,2), поскольку она является ближайшей согласно алгоритму Ли. Хотя эта точка и не элемент сети, но в ее буфере содержится информация, что она подключена к искомой ячейке. Поэтому эта ячейка принимается за конечную, и буфер всего пути изменяется (рис. 4,*б*), поскольку к текущему пути добавлен новый элемент.



а) *б*)

Рис. 4. Схема соединения между элементами *A*, *B*, *D*

3. Остальные соединения находятся аналогичным образом.

Следует отметить, что модифицированный алгоритм Ли был реализован программно на языке C/C# и был использован при построении человеко-машинного интерфейса системы трансляции и визуализации схем электроподстанций, построенных в соответствии со стандартом IEC 61850, в виде функциональных блоков в системе NxtStudio [5].

В ходе выполнения данной работы был разработан модифицированный алгоритм Ли, описаны его отличия и представлен пример его работы. В дальнейшем планируется использование этого алгоритма при разработке реальной системы Smart Grid.

Библиографический список

1. Gharavi, H. Guest Editorial Special Section on Cyber-Physical Systems and Security for Smart Grid / H. Gharavi, H. Chen, C. Wietfeld // IEEE Transactions on Smart Grid. – 2015. – Vol. 6 (5). – P. 2405–2408.

2. IEC 61850 – Communication networks and systems for power utility automation, IEC Standard, Part 6: Configuration language for communication in electrical substations related to IEDs – Ed 2, 2009.

3. Christensen, J. H. IEC 61499 Architecture, Engineering Methodologies and Software Tools / J. H. Christensen // Knowledge and Technology Integration in Production and Services: Balancing Knowledge and Technology in Product and Service Life Cycle. – Boston, MA: Springer US, 2002. – P. 221–228.

4. Lee, C. Y. An Algorithm for Path Connections and Its Applications / C. Y. Lee // IRE Transactions on Electronic Computers. – 1961. – Vol. EC-10, № 3. – P. 346–365.

5. nxtStudio. – URL: – http://www.nxtcontrol.com/

И. В. Сенокосов, А. С. Войнов

Пензенский государственный университет, Пенза, Россия

ПРОГРАММНАЯ МОДЕЛЬ СИСТЕМЫ СБОРКИ СЛОЖНЫХ ИЗДЕЛИЙ НА ОСНОВЕ МАГАЗИННЫХ АВТОМАТОВ-ПРЕОБРАЗОВАТЕЛЕЙ В СРЕДЕ NXTSTUDIO

Предложена программная модель системы сборки сложных изделий на основе магазинных автоматов-преобразователей, реализованная на основе функциональных блоков стандарта МЭК 61499 в среде NxtStudio. Ее функционирование проиллюстрировано на конкретном примере.

В настоящее время важной составной частью производственных систем наряду с системами выборки являются системы сборки сложных изделий из разнотипных составных

частей. Для ряда производственных сборочных систем важной является последовательность компонентов, подаваемой на их вход, из которых конструируется сложное изделие. Кроме того, для сборочного производства вполне актуальной может стать задача распознавания структуры последовательности движущихся по конвейеру деталей для принятия решения о дальнейшем ходе техпроцесса сборки.

В работе [1] было предложено использование магазинных автоматов для проектирования систем сортировки и сборки деталей. В данной работе предлагается программная модель сборки изделий на основе магазинного автомата-преобразователя (МА) [2]. В модели используется концепция технологического процесса сборки изделий, управляемого потоком деталей и жетонов. При этом изделие собирается из поступающих (например, по конвейеру) деталей, а технологические операции описываются (управляющими) жетонами. Жетоны используются для управления сборкой изделий, но при этом сами непосредственно не входят в изделие.

В качестве примера построения модели сборки изделий были взяты три типа деталей: черная (*black, b*), оранжевая (*orange, o*) и серая (*gray, g*). Также были определены четыре основных операции сборки (в скобках указаны сокращения, используемы в данной работе): «приклеивание» детали слева (*left, l*) – действие *d1*, «приклеивание» детали справа (*right, r*) – действие *d2*, «приклеивание» детали сверху (*up, u*) – действие *d3* и «приклеивание» детали снизу (*down, d*) – действие *d4*. В качестве геометрической (координатной) основы для выполнения операций принимается последняя деталь, установленная на изделие. При таком подходе необходимо предусмотреть решение «проблемы занятой стороны», такого случая при котором необходимая сторона уже занята другой деталью. Например, если к одной детали «приклеить» вторую сверху, то координатная основа сместится и ко второй детали «приклеить» снизу третью не получится. Для решения данной проблемы достаточно пропускать «проблемную» операцию сборки. Кроме того, деталь можно приклеивать на первое свободное место по необходимому направлению (для примера в данной работе третью деталь можно приклеить снизу к первой).

Рассмотрим работу модели сборки для построения изделия одинаковой длины и высоты, напоминающего бумеранг. При этом МА использует только две технологические операции *left* и *up*. Данный МА и примеры изготавливаемых им изделий приведены на рис. 1. Необходимо отметить, что каждый переход в графическом представлении МА описывается кортежем, включающим 1) входной символ; 2) магазинный символ вершины стека; 3) замещающую последовательность магазинных символов и 4) действие (необязательно).

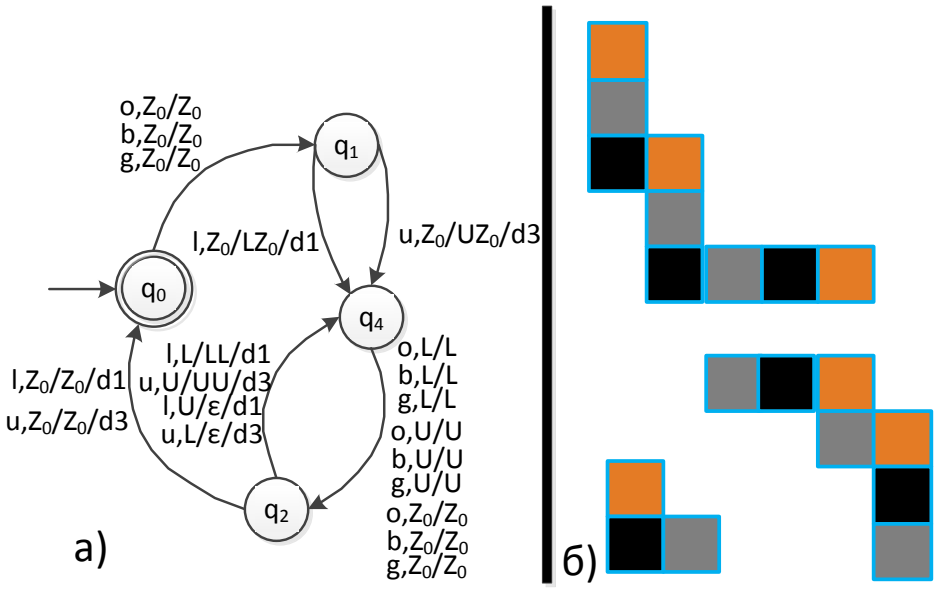


Рис. 1. *а* – МА, реализующий сборку изделий;
б – примеры готовых сконструированных изделий

Как видно из рис. 1,*а*, магазинными символами являются *L, U, Z*$_0$. Начальный магазинный символ – Z_0. Начальное состояние – q_0. Конечное состояние – q_0. Одинаковые начальное и конечное состояния позволяют продолжить функционирование МА, не сбрасывая его.

Рассмотрим замкнутую имитационную модель системы сборки изделий (рис. 2), реализованную на основе функциональных блоков стандарта МЭК 61499 в системе nxtStudio [3] в соответствии с шаблоном MVC. Сама система состоит из выталкивателя; трех датчиков для распознавания типа детали или жетона; двух датчиков определения положения выталкивателя; емкостного датчика, определяющего, была ли вытолкнута деталь/жетон в шахту; заслонки для забора деталей/жетонов из шахты; приемника жетонов; места для построения изделия и светодиодного индикатора, сигнализирующего об окончании сборки. В человеко-машинном интерфейсе также отображается стек управляющего МА для контроля его работы.

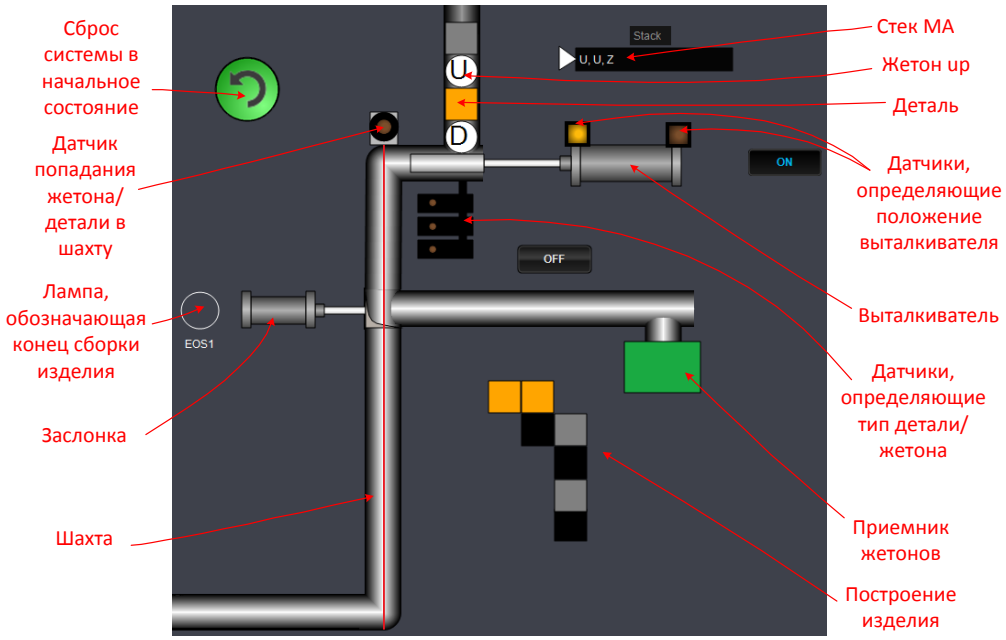


Рис. 2. Система сборки изделий на основе МА (интерфейс пользователя в nxtStudio)

Имитационная модель системы сборки изделий, включающая систему управления на основе функциональных блоков стандарта МЭК 61499 и визуальную модель оборудования, работает следующим образом: сначала генерируется деталь/жетон определенного цвета/типа, который распознается с помощью датчиков и попадает на анализ управляющему МА. Если деталь/жетон подходит, то сигнал «выдвинуть заслонку» передается в модель сортировки, и, таким образом, деталь/жетон попадает в боковую шахту. Если прибыл жетон, то он попадает в приемник жетонов и сообщает направление «приклеивания» детали. Деталь же продолжает двигаться и «приклеивается» к нужному месту. В противном случае деталь/жетон движутся дальше по шахте. Когда заслонка выдвинута, но деталь/жетон не подходит, то она задвигается и пропускает деталь/жетон, падающие вниз по шахте.

В представленной работе представлена модель сборки изделий, управляемая МА-преобразователем в среде nxtStudio. Направлением дальнейших исследований является усложнение данной модели, а также разработка системы сборки 3D-изделий.

Библиографический список

1. Дубинин, В. Н. Использование моделей магазинных автоматов в проектировании технологических процессов сортировки и сборки изделий / В. Н. Дубинин, И. В. Сенокосов, Л. П. Климкина, В. В. Вяткин // Современные технологии в науке и образовании : сб. тр. II Междунар. науч.-техн. и науч.-методической конф. (СТНО-2017). – Рязань, 2017. – Т. 1. – С. 10–14.

2. Хопкрофт, Дж. Введение в теорию автоматов, языков и вычислений / Дж. Хопкрофт, Р. Мотвани, Дж. Ульман. – М. : Вильямс, 2002. – 528 с.

3. Сайт nxtStudio (nxtControl). – URL: http://www.nxtcontrol.com/

Д. А. Солопов, Е. С. Мордасова, И. В. Калистратова

Тамбовский государственный технический университет, Тамбов, Россия

ПОДХОД К АВТОМАТИЗИРОВАННОМУ ВЫБОРУ ТИПА БЛОКА ШТАМПА1

Предложен способ автоматизированного выбора штампа вырубки и пробивки, базирующийся на использовании знаний предметной области, представленных в виде правил (продукций) вида «Если ..., то ...».

Штамповка является одним из самых эффективных методов обработки металлов давлением. Готовое изделие, полученное штамповкой, не требует дополнительной механической обработки, что значительно уменьшает его себестоимость.

Оборудование холодной штамповки состоит преимущественно из стандартных типов блоков штампов и деталей. Основными узлами штампа являются комплект верхних и нижних плит и связующие их направляющие устройства. Используемы типы стандартных блоков штампов достаточно разнообразны. На рис. 1 представлены блоки штампов, соответствующие стандартам ГОСТ 13125–83 «Блоки штампов с задним расположением направляющих узлов скольжения» и ГОСТ 13124–83 Блоки штампов с диагональным расположением направляющих узлов скольжения».



а) *б*)

Рис. 1. Примеры блока штампов: *а* – ГОСТ 13125–83 «Блоки штампов с задним расположением направляющих узлов скольжения»; *б*) ГОСТ 13124–83 Блоки штампов с диагональным расположением направляющих узлов скольжения»

Процесс разработки штампа включает в себя следующие этапы:

– выбор типа блока;

– расчет основных параметров;

– разработка конструкции отдельных деталей.

Выбор типа блока штампа осуществляется на основании анализа геометрии и точности штампуемых изделий. Предлагается осуществлять выбор типа блока штампа на основании анализа знаний предметной области, представленных в виде правил (продукций) вида «Если ..., то ...».

[1] Работа выполнена под руководством профессора кафедры КИСМ ТГТУ В. Г. Мокрозуба.

Правило 1. Если операции штамповки разделительные и повышенные требования к точности штампуемого изделия, то тип блока штампа по ГОСТ 13124–83 «Блоки штампов с диагональным расположением направляющих узлов скольжения».

Правило 2. Если операции штамповки несложные гибочные, то тип блока штампа по ГОСТ 13125–83 «Блоки штампов с задним расположением направляющих узлов скольжения».

Правило 3. Если заготовка – полоса и и повышенные требования к точности штампуемого изделия, то тип блока штампа по ГОСТ 13124–83 Блоки штампов с диагональным расположением направляющих узлов скольжения».

Предложенный способ определения типа блока штампа используется при разработке системы автоматизированного проектирования технологического оборудования в Тамбовском государственном техническом университете [1–6].

Библиографический список

1. Новые информационные технологии в открытом инженерном образовании / Е. Н. Малыгин, С. В. Карпушкин, М. Н. Краснянский, В. Г. Мокрозуб, А. Б. Борисенко. – М. : Машиностроение-1, 2003. – 124 с.

2. Разработка прототипа виртуальной модели учебно-материальных ресурсов университета химико-технологического профиля / В. А. Немтинов, В. В. Юханов, Е. Н. Малыгин, С. В. Карпушкин, С. Я. Егоров, В. Г. Мокрозуб, А. Б. Борисенко, Ю. В. Немтинова // Вопросы современной науки и практики. Университет им. В. И. Вернадского. – 2013. – № 3 (47). – С. 321–330.

3. Мокрозуб, В. Г. Структура интеллектуальной автоматизированной системы механических расчетов технологического оборудования / В. Г. Мокрозуб, А. В. Мокрозуб, А. А. Чуриков // Вопросы современной науки и практики. Университет им. В. И. Вернадского. – 2015. – № 4 (58). – С. 187–195.

4. Мокрозуб, В. Г. Разработка интеллектуальных информационных систем автоматизированного проектирования технологического оборудования : учеб. пособие / В. Г. Мокрозуб. – Тамбов : Тамбовский ГТУ, 2008. – 80 с.

5. Мокрозуб, В. Г. Структура информационно-логической модели кожухотрубчатых теплообменников / В. Г. Мокрозуб, С. В. Морозов // Вестник Тамбовского государственного технического университета. – 2013. – Т. 19, № 3. – С. 518–526.

6. Мокрозуб, В. Г. Постановка задачи разработки математического и информационного обеспечения процесса проектирования многоассортиментных химических производств / В. Г. Мокрозуб, Е. Н. Малыгин, С. В. Карпушкин // Вестник Тамбовского государственного технического университета. – 2017. – Т. 23, № 2. – С. 252–264.

Ю. С. Семерич, Л. В. Тельнова

Пензенский государственный университет, Пенза, Россия

ПРИМЕНЕНИЕ R-ФУНКЦИЙ ДЛЯ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ ГЕОМЕТРИЧЕСКИХ ОБЪЕКТОВ С ПЕРФОРАЦИЕЙ

В работе развивается конструктивный аппарат теории R-функций для геометрического моделирования объектов с перфорацией. Основа предложенного подхода для проектирования таких областей – способ учета симметрии трансляционного типа. Получены уравнения для различных областей, подлежащих трансляции.

Автоматизация построения инженерных конструкций и сооружений из геометрических областей – это задача, которая часто возникает в компьютерном и математическом моделировании, в работе с приложениями компьютерной графики и численного анализа.

На деле разработчик сталкивается с моделями сложных форм, следовательно, реализация областей на практике затруднена. Стоит отметить, что точность получившегося численного результата будет зависеть от качества реализации геометрической модели.

Отметим, что в области авиастроения и аэрокосмических исследований разрабатываются звукопоглощающие конструкции авиационных двигателей, конструктивной особенностью которых является то, что в поверхностном слое оболочки выполнены сквозные отверстия (перфорация), в машиностроении часто используют щиты из перфорированного металла, которые предназначены для защиты станков и обеспечивают безопасность деятельности работников предприятия.

На сегодняшний день, существуют разнообразные варианты моделирования областей двухмерной формы и 3D-объектов. Их классификацию производят следующим образом:

1. Описание поверхности параметрически, когда рассматриваются сегменты границы, которые образуют геометрическую фигуру.

2. Геометрическое твердотельное моделирование, как правило, геометрическая трехмерная модель воссоздается по трем ее проекциям.

3. Композиция топологической модели заданной конструкции, как совокупность базовых фигур (линии) и логических операций над ними, таких как пересечение, объединение и вычитание над ними.

У каждого из перечисленных выше видов моделирования есть слабые стороны. Первый подход – трудоемкий процесс для построения объектов сложной формы. Второй – не для всех случаев позволяет получить пригодную модель для дальнейшего применения. Третий случай не всегда позволяет описывать необходимые модели в работе, из-за ограниченности базовых фигур.

Можно сделать вывод о том, что необходим универсальный метод для автоматизации построения геометрических фигур, и то что, этот вопрос на сегодняшний день не потерял своей актуальности. Возможным вариантом решения данной проблемы является применение конструктивной теории *R*-функций, разработанной В. Л. Рвачевым [1].

R-функции – логически изменяемые действительные функции, у которых знак зависит от знака их аргументов. Рассмотрим функции:

$$W_1 = xyz \tag{1}$$

$$W_2 = x + y + \sqrt{x^2 + y^2 + xy} \tag{2}$$

В табл. 1 отражена зависимость знака функций W_1 и W_2 от знака их аргументов.

Таблица 1

Зависимость знака функций W_1 и W_2 от знака их аргументов

x	y	z	W_1	W_2
–	–	–	–	–
–	–	+	+	–
–	+	–	+	+
–	+	+	–	+
+	–	–	+	+
+	–	+	–	+
+	+	–	–	+
+	+	+	+	+

R-функции разрешают произвести описание границы геометрического объекта на аналитическом уровне. Данный процесс состоит в задании функции, которая принимает внутри области положительные значения, на границы области нулевые значения, а вне области – отрицательные. Построение *R*-функции осуществляется по правилам. Первый этап – задание опорных областей, являющихся базовыми фигурами. Второй этап – построение необходимой модели, состоящей из комбинации областей, полученных на первом этапе, и *R*-функций.

Рассмотрим полную систему *R*-функций – это система $\mathfrak{R}_\alpha$ [2]:

$$C \equiv \text{const},$$
$$\overline{x} \equiv -x,$$

$$x_1 \Lambda_\alpha x_2 \equiv \frac{1}{1+\alpha}\left(x_1 + x_2 - \sqrt{x_1^2 + x_2^2 - 2\alpha x_1 x_2}\right), \tag{3}$$

$$x_1 \mathrm{v}_\alpha x_2 \equiv \frac{1}{1+\alpha}\left(x_1 + x_2 + \sqrt{x_1^2 + x_2^2 - 2\alpha x_1 x_2}\right).$$

где Λ_α – *R*-конъюнкция, v_α – *R*-дизъюнкция, а параметр $\alpha(x,y)$ выбирается из следующего условия:

$$-1 < \alpha(x,y) < 1, \alpha(x,y) \equiv \alpha(y,x) \equiv \alpha(-x,y) \equiv \alpha(-y,x).$$

Если $\alpha \equiv 0$, то получим систему $\mathfrak{R}_0$, как раз ее чаще всего используют на практике: $C \equiv \text{const},$

$$\overline{x} \equiv -x,$$

$$x_1 \Lambda_0 x_2 \equiv \left(x_1 + x_2 - \sqrt{x_1^2 + x_2^2}\right), \tag{4}$$

$$x_1 \mathrm{v}_0 x_2 \equiv \left(x_1 + x_2 + \sqrt{x_1^2 + x_2^2}\right).$$

Описанный подход получил развитие и позволил описать геометрические области, обладающие симметрией трансляционного типа. Считают, что объект *L* обладает ей в направлении некоторого вектора $\vec{a}(a_1, a_2, a_3)$, если при преобразовании

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} \leftarrow \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix}, \tag{5}$$

он переходит сам в себя.

В работе рассматривались случаи, когда область, подлежащая трансляции, обладала симметрией относительно оси ординат, не обладала такой симметрией, а также когда две соседние области не могли быть отделены одна от другой вертикальной полосой.

Для построения уравнений границ областей, использовались частичные суммы ряда Фурье по синусам

$$\mu(x,h) = \frac{4h}{\pi^2} \sum_{i=1}^{\infty} \frac{(-1)^{i+1}}{(2i-1)^2} \sin\left[\frac{(2i-1)x\pi}{h}\right], \tag{6}$$

где h – шаг трансляции.

Рассмотрим случай, когда область σ_0 симметрична осям координат. Зададим ее в таком виде:

$$\sigma_0 = \omega_1 \mathrm{v}_\alpha \omega_2, \tag{7}$$

где $\omega_1 = \frac{1}{2a}\left(a^2 - x^2\right)$, $\omega_2 = \frac{1}{2a}\left(a^2 - y^2\right)$.

Опишем трансляцию опорной области вдоль оси OX[3]:

$$\omega_1(x,y) \equiv \sigma_0\left(\mu_n(x,h), y\right) = 0. \tag{8}$$

Используя метод *R*-функций, была получена формула для перфорированной области $\omega_1(x,y)$ вдоль оси *OY*:

$$\omega_2(x,y) \equiv \omega_1\left(x,\mu(y)\right). \tag{9}$$

На рис. 1 приведена область $\omega_2(x,y)$.



а) *б*)

Рис. 1. Область $\omega_2(x,y)$: *а*) схематическое изображение области $\omega_2(x,y)$;
б) картина линий уровня функции $\omega_2(x,y)$

Перейдем к рассмотрению случая, когда опорная область σ_1 не обладает симметрией относительно осей координат. Трансляционную область зададим так:

$$\sigma_1 = \tau_1 \vee_0 \tau_2, \tag{10}$$

где $\tau_1 = (\overline{\eta_1} \vee_0 \overline{\eta_2}) \vee_0 \left(\overline{\eta_3} \vee_0 \overline{\eta_4}\right)$, $\eta_1 = -y + x - 4$, $\eta_2 = y - x - 4$ $\eta_3 = -y - x - 4$, $\eta_4 = y + x - 4$,
$\tau_2 = R^2 - (x-a)^2 - (y-b)^2$.

Если воспользоваться формулой (8) для трансляции σ_1 относительно оси *OX*, то появятся зеркально отраженные области. Поэтому уравнение, описывающее трансляцию σ_1 вдоль оси *OX*, будет выглядеть так:

$$\omega_4(x,y) \equiv \left\{\sigma_1\left(\mu_n(x,h),y\right)\Lambda_\alpha \cos\left(\frac{x\pi}{h}\right)\right\} \vee_\alpha \left\{\sigma_1\left(\mu_n(x-h,h),y\right)\Lambda_\alpha \cos\left(\frac{(x-h)\pi}{h}\right)\right\} = 0. \tag{11}$$

Затем используем метод R-функций для трансляции области $\omega_4(x,y)$ вдоль оси *OY*:

$$\omega_5(x,y) \equiv \omega_4\left(x,\mu(y,h)\right) = 0. \tag{12}$$

На рис. 1 приведена область $\omega_5(x,y)$.

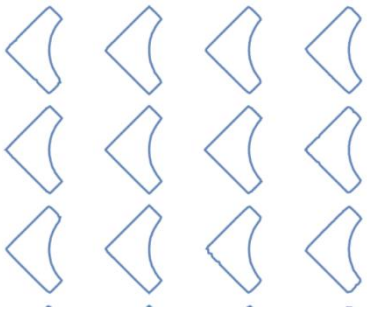


Рис. 2. Область $\omega_5(x,y)$

В качестве третьей опорной области выберем область, у которой при трансляции соседние области не могут быть отделены друг от друга вертикальной полосой. К ней, не применима формула (2). Поэтому уравнение, описывающее трансляцию σ_2 вдоль оси *OX* будет выглядеть так:

$$\omega_7(x,y)\equiv\left\{\left[\sigma_2\left(\mu_n(x,2h),y\right)\Lambda_\alpha\cos\left(\frac{x\pi}{2h}\right)\right]\mathrm{v}_\alpha\left[\sigma_2\left(\mu_n(x-2h,2h),y\right)\Lambda_\alpha\cos\left(\frac{(x-2h)\pi}{2h}\right)\right]\right\}\mathrm{v}_\alpha$$

$$\left\{\left[\sigma_2\left(\mu_n(x-p,2h),y\right)\Lambda_\alpha\cos\left(\frac{(y-h)\pi}{2h}\right)\right]\mathrm{v}_\alpha\left[\sigma_2\left(\mu_n(x-3h,2h),y\right)\Lambda_\alpha\cos\left(\frac{(y-3h)\pi}{2h}\right)\right]\right\}=0. \quad (13)$$

Затем используем метод *R*-функций для трансляции области $\omega_7(x,y)$ вдоль оси *OY*:

$$\omega_8(x,y)\equiv\left\{\left[\omega_7\left(x,\mu_n(y,2h)\right)\Lambda_\alpha\cos\left(\frac{y\pi}{2h}\right)\right]\mathrm{v}_\alpha\left[\omega_7\left(x,\mu_n(y-2h,2h)\right)\Lambda_\alpha\cos\left(\frac{(y-2h)\pi}{2h}\right)\right]\right\}\mathrm{v}_\alpha$$

$$\left\{\left[\omega_7\left(x,\mu_n(x-h,2h),\right)\Lambda_\alpha\cos\left(\frac{(y-h)\pi}{2h}\right)\right]\mathrm{v}_\alpha\left[\omega_7\left(x,\mu_n(x-3h,2h)\right)\Lambda_\alpha\cos\left(\frac{(y-3h)\pi}{2h}\right)\right]\right\}=0. \quad (14)$$

Таким образом, использование конструктивной теории *R*-функций помогает эффективно и достаточно легко описывать перфорированные объекты. Применение использованного подхода для трехмерного случая разрешит увеличить качество реальных геометрических конструкций.

Для построения уравнений границ геометрических областей с перфорацией достаточно знать информацию об уравнении только одной области и типе перфорации. Процесс построения областей трех видов был автоматизирован. Программная реализация, полученных формул, производилась с использованием пакета численной математики Wolfram Mathematica 10.3 и в программирующей среде Поле.

Библиографический список

1. Рвачев, В. Л. Теория R-функций и некоторые ее приложения / В. Л. Рвачев. – Киев : Наук. думка, 1982. – 506 с.

2. Толок, А. В. Построение симметричных функций для симметричных чертежей / А. В. Толок, Ю. С. Семерич, Т. И. Шейко // Вестник ЗГУ. – 2001. – № 2. – С. 83–97.

3. Семерич Ю. С. Развитие конструктивных средств учета геометрической симметрии в методе *R*-функций / Ю. С. Семерич // Аналитические и численные методы моделирования естественнонаучных и социальных проблем : сб. ст. II Междунар. науч.-техн. конф. (Пенза, Россия, 4–5 сентября 2007 г.). – Пенза, 2017. – С. 36–39.

Е. Н. Салмов

ООО «Маяктрансэнерго», Пенза, Россия

МОДЕЛИРОВАНИЕ УПРАВЛЯЕМЫХ ВЕНТИЛЬНО-ЭЛЕКТРОМЕХАНИЧЕСКИХ СИСТЕМ С ПЕРЕМЕННОЙ НАГРУЗКОЙ

Рассмотрен метод моделирования управляемых вентильно-электромеханических систем с переменной нагрузкой, построенных на базе непосредственных преобразователей электроэнергии и асинхронных машин. Рассмотрено получение экспериментальных данных. Рассмотренный подход отличается использованием минимального количества моделируемых состояний.

Отличием рассмотренного в статье подхода к моделированию электропривода с переменной нагрузкой является исследование всех возможных режимов системы «непосредственный преобразователь электроэнергии – асинхронный двигатель» с произвольной структурой вентильного преобразователя на базе трех: симметричный режим (переключа-

ющие элементы (ПЭ) включены во всех трех фазах), двухфазный режим (ПЭ выключены в одной из трех фаз) и режим выбега (ПЭ выключены во всех фазах) [1–6]. Это позволяет достичь значительного упрощения моделирования при сохранении точности результатов и с учетом переменного характера нагрузки.

Математическое моделирование выполняется в двухфазной колеблющейся системе координат. При этом отключенная фаза совмещается с осью α и число моделируемых состояний системы уменьшается до трех. Входное напряжение электродвигателя в двухфазной системе координат зададим в векторной форме [1, 2]:

$$\begin{bmatrix} U_{1\alpha} \\ U_{1\beta} \end{bmatrix} = U_m \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{3}} \end{bmatrix} \times \begin{bmatrix} H^1_{1a}[t] & 0 & 0 \\ 0 & H^1_{1b}[t] & 0 \\ 0 & 0 & H^1_{1c}[t] \end{bmatrix} \times \begin{bmatrix} \sin(\omega 0t - H^2_A[t]\Delta\varphi) \\ \sin(\omega 0t - H^2_B[t]\Delta\varphi) \\ \sin(\omega 0t - H^2_C[t]\Delta\varphi) \end{bmatrix} \quad (1)$$

где $U_{1\alpha}, U_{1\beta}$ – проекции обобщенного вектора напряжения статора на оси α и β; $H^2_A, H^2_B, H^2_C, H^1_{1a}, H^1_{1b}, H^1_{1c}$ – переключающие функции.

Для исследования асинхронного электродвигателя в симметричном режиме при общепринятых допущениях строится следующая математическая модель в двухфазной системе координат α,β [2, 5, 6]:

$$\begin{bmatrix} \frac{d\psi_{1\alpha}}{dt} \\ \frac{d\psi_{1\beta}}{dt} \\ \frac{d\psi_{2\alpha}}{dt} \\ \frac{d\psi_{2\beta}}{dt} \end{bmatrix} = \begin{bmatrix} -R_1F_2 & 0 & R_1F_3 & 0 \\ 0 & -R_1F_2 & 0 & R_1F_3 \\ R_2F_3 & 0 & -R_2F_1 & -\omega_Э \\ 0 & R_2F_3 & -\omega_Э & -R_2F_1 \end{bmatrix} \times \begin{bmatrix} \psi_{1\alpha} \\ \psi_{1\beta} \\ \psi_{2\alpha} \\ \psi_{2\beta} \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} U_{1\alpha} \\ U_{1\beta} \\ U_{2\alpha} \\ U_{2\beta} \end{bmatrix};$$

$$\begin{bmatrix} i_{1\alpha} \\ i_{1\beta} \\ i_{2\alpha} \\ i_{2\beta} \end{bmatrix} = \begin{bmatrix} F_2 & 0 & -F_3 & 0 \\ 0 & F_2 & 0 & -F_3 \\ -F_3 & 0 & F_1 & 0 \\ 0 & -F_3 & 0 & F_1 \end{bmatrix} \times \begin{bmatrix} \psi_{1\alpha} \\ \psi_{1\beta} \\ \psi_{2\alpha} \\ \psi_{2\beta} \end{bmatrix};$$

$$M = \frac{3}{2} p_П F_3 \cdot \begin{bmatrix} -\psi_{1\alpha} & \psi_{1\beta} \end{bmatrix} \cdot \begin{bmatrix} \psi_{2\beta} \\ \psi_{2\alpha} \end{bmatrix}; \quad (2)$$

$$\begin{bmatrix} i_{m\alpha} \\ i_{m\beta} \end{bmatrix} = \begin{bmatrix} F_2 - F_3 & 0 & F_1 - F_3 & 0 \\ 0 & F_2 - F_3 & 0 & F_1 - F_3 \end{bmatrix} \times \begin{bmatrix} \psi_{1\alpha} \\ \psi_{1\beta} \\ \psi_{2\alpha} \\ \psi_{2\beta} \end{bmatrix};$$

$$\frac{d\omega}{dt} = \frac{1}{J}\left[M - M_c \cdot \text{sign}(\omega) \cdot \xi \cdot (1 - \sin(\xi t) \cdot \text{sign}(\sin(\xi t)))\right].$$

где $\psi_{1\alpha}$, $\psi_{1\beta}$, $\psi_{2\alpha}$, $\psi_{2\beta}$ и $i_{1\alpha}$, $i_{1\beta}$ – проекции обобщенного вектора потокосцепления статора, потокосцепления ротора и тока статора; $i_{m\alpha}$, $i_{m\beta}$ – проекции обобщенного вектора

намагничивающего тока; i_m – модуль вектора намагничивающего тока; R_1, R_2 – активные сопротивления статорной и роторной цепей; $p_П$ – число пар полюсов; ω – скорость ротора; ω_0 – синхронная скорость; M – электромагнитный момент; x_1, x_2, x_m – индуктивные сопротивления статора, ротора и намагничивающего контура; i_a, i_b, i_c – токи в трехфазной системе координат; M_c, J – момент сопротивления и момент инерции; ξ – коэффициент изменения нагрузки.

Значения коэффициентов в (2) определяются по формулам [2]:

$$F_1 = \frac{\omega}{x_m(x_1 + x_2) + x_1 x_2} \cdot (x_m + x_1); \quad F_2 = \frac{\omega}{x_m(x_1 + x_2) + x_1 x_2} \cdot (x_m + x_2);$$

$$F_3 = \frac{\omega}{x_m(x_1 + x_2) + x_1 x_2} \cdot x_m .$$

Для проверки адекватности математического моделирования были собраны экспериментальные данные по работе электропривода отрубного агрегата Martin с вариатором и обгонной муфтой, режимы которых задаются в зависимости от размера отруба. Электропривода экспериментальной установки построен на базе асинхронного двигателя и вентильного преобразователя мощностью 15 кВт. Для определения нагрузок в различных режимах работы отруба были проведены замеры тока нагрузки и питающего тока на одной из вводных фаз преобразователя. Для выполнения замеров использовался высокоскоростной регистратор аналоговых сигналов ЛА-5. Для размеров отруба 1000 мм, 2000 мм и 3000 мм были проведены замеры на скоростях от 10 до 70 м/мин с шагов в 10 м/мин (замер на каждой скорости выполнялся в течение одной минуты). Для получения данных работы отруба в регистраторе использовались два канала с частотой выборки 16000 Гц по каждому каналу. Таким образом получены по 10080000 измерений по каждому из двух рассмотренных параметров. Результаты измерений тока нагрузки на размерах 1000 мм и 3000 мм показаны на рис. 1 и 2 соответственно (значение 5 соответствует току нагрузки 23 Ампера). Пиковые значения соответствуют максимальной нагрузке в момент отруба.

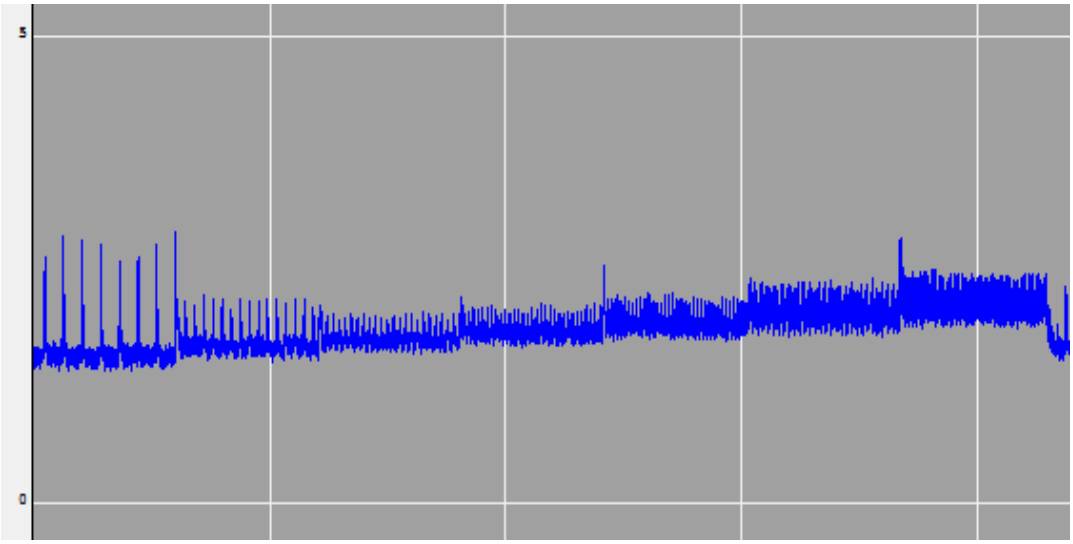


Рис. 1. Экспериментальные данные для размера 1000 мм

Отклонение результатов моделирования от полученных экспериментальных данных не превышает 9 %, что подтверждает адекватность математического моделирования. Разработанный подход, полученные формулы (1) – (2) и программный комплекс математического моделирования [4] являются основой для исследования как индивидуального, так и многодвигательного привода с асинхронными электродвигателями и переменной нагрузкой.

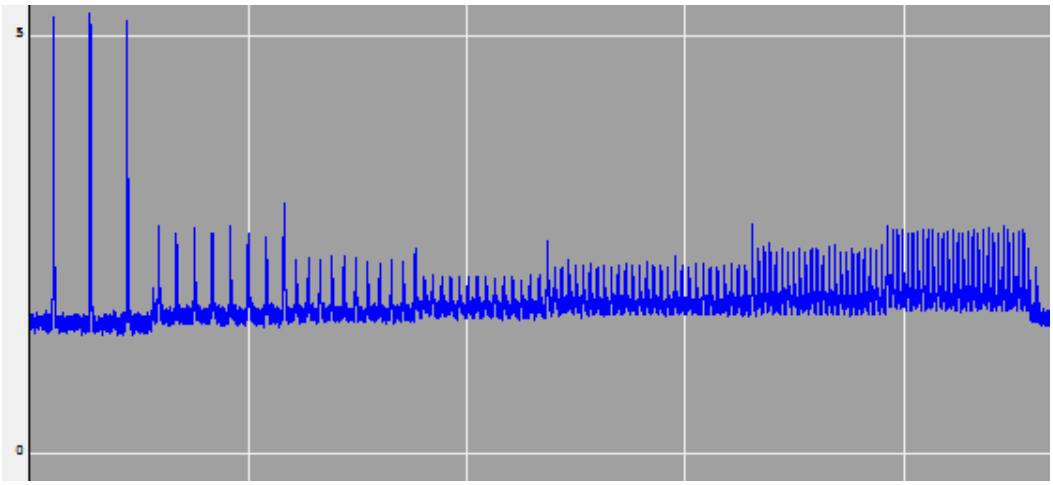


Рис. 2. Экспериментальные данные для размера 3000 мм

Библиографический список

1. Прошин, И. А. Управление в вентильно-электромеханических системах. Кн. 1. Управление непосредственным преобразованием электрической энергии / И. А. Прошин. – Пенза : Изд-во ПТИ, 2003. – 333 с.

2. Прошин, И. А. Управление в вентильно-электромеханических системах. Кн. 2. Математическое моделирование вентильно-электромеханических систем / И. А. Прошин. – Пенза : Изд-во ПТИ, 2003. – 307 с.

3. Салмов, Е. Н. Алгоритм комбинированного управления и метод математического моделирования асинхронного электропривода / И. А. Прошин, М. И. Вольников, Е. Н. Салмов // Нива Поволжья. – 2014. – № 4 (33) – С. 102–109.

4. Салмов, Е. Н. Интегрированный комплекс компьютерно-имитационного моделирования электромеханических систем / Е. Н. Салмов, Р. Д. Прошина, Е. С. Прошина // XXI век: итоги прошлого и проблемы настоящего плюс. – 2013. – № 12 (16). – С. 95–101.

5. Салмов, Е. Н. Математическое моделирование системы электропривода ролико-опор / Е. Н. Салмов, И. А. Прошин, Д. А. Кузнецов // XXI век: итоги прошлого и проблемы настоящего плюс. – 2015. – № 05 (27). – С. 285–293.

6. Прошин, И. А. Теоретические основы моделирования управляемых вентильно-электромеханических систем с непосредственными преобразователями электрической энергии / И. А. Прошин // Информационные технологии в проектировании и производстве. – 2000. – № 4. – С. 65–70.

С. А. Акопян, В. Н. Акопян, В. А. Мокрозуб

Тамбовский государственный технический университет, Тамбов, Россия

БАЗА ЗНАНИЙ СИСТЕМЫ АВТОМАТИЗИРОВАННОГО СОСТАВЛЕНИЯ ТЕХНОЛОГИИ ИЗГОТОВЛЕНИЯ ВАЛОВ[1]

Представлена структура базы данных для хранения онтологии, предназначенной для разработки технологии изготовления валов.

Задача определения подходящего вида механической обработки решается, как на этапе разработки технологической документации при проектировании новых изделий, так и на этапе планирования необходимых ресурсов для выполнения производственного задания.

[1] Работа выполнена под руководством профессора кафедры КИСМ ТГТУ В. Г. Мокрозуба.

Механическая обработка включает в себя вид обработки (токарная, фрезерная и т.д.) и ее параметры (черновое точение, чистовое растачивание и т.д.), которые зависят от ответственности детали, обрабатываемого элемента и других параметров.

Одним из способов автоматизации поддержки принятия решений при определении подходящего вида механической обработки при обработке деталей является использование онтологии предметной области.

Формально онтология, предназначенная для определения подходящего вида механической обработки при обработке определенного элемента детали, определяется как $OP = \langle P, T, G \rangle$, где P – таксономия (дерево) видов механической обработки (токарная, черновое точение); T – таксономия (дерево) видов деталей и обрабатываемых элементов (вал, ступень); G – правила, связывающие вершины дерева видов механической обработки с вершинами дерева видов деталей и обрабатываемых.

Онтология такого рода, может быть реализована в формате базы данных (рис. 1). Она состоит из девяти таблиц.

1. Виды обрабатываемых элементов. В этой таблице перечислены все возможные конструкционные элементы, деталей типа «Вал» (отверстие продольное, ступень, резьба и др.).
2. Шероховатость обработки Ra. В ней перечислены диапазоны шероховатостей по Ra.
3. Шероховатость обработки Rz. Здесь перечислены диапазоны шероховатостей по Rz.
4. Квалитет обработки. В этой таблице перечислены квалитеты точности обработки.
5. Операции. В ней перечислены все возможные механические операции (токарное точение, токарное растачивание, сверление, зенкерование).
6. Свойства операции. Здесь перечислены свойства операции с разными значениями (чистота обработки: чистовая, чистота обработки: черновая и т.д.).
7. Возможные варианты элементов. В этой таблице перечислены возможные совпадения значений из таблиц 1, 2, 3 и 4. С исключением несуществующих вариаций.
8. Возможные варианты операций. В ней перечислены возможные совпадения значений из таблиц 5 и 6. С исключением несуществующих вариаций.
9. Правила. Здесь находятся связи, связывающие определенную вариацию элементов с определенной вариацией механической обработки.
10. Запрос. Это изначально пустая таблица, предназначенная для сбора и хранения информации, введенной пользователем.
11. Запрос код. Данная таблица предназначена для хранения информации из таблицы 10, отличие лишь в том, что значения ячеек изменяются на соответствующие им значения «код» из таблиц 1, 2, 3 и 4.
12. Результат. В эту таблицу обработчик записывает значения таблиц 5 и 6, соответствующие значениям ячейки «код» таблицы 8, полученной в результате совпадения значений таблиц 7 и 8, в таблице 9.

Пользователь вводит необходимые параметры, которые отправляются на хранение в таблицу 10 (элемента – продольное отверстие; шероховатость_Ra – 5; шероховатость_Rz – нет; квалитет – 7).

Далее обработчик находит соответствующие значения ячеек «код» из таблиц 1, 2, 3 и 4, и записывает их в таблицу 11 (код_элемента – соответствующий продольному отверстию; код_шероховатости_Ra – соответствующий шероховатости 5; код_шероховатости_Rz – нет; код_квалитет – соответствующий квалитету 7)

Потом обработчик сравнивает значения таблиц 11 и 7. И после нахождения совпадения, записывает значение соответствующего «кода» строки таблицы 7 в последнюю ячейку таблицы 11.

В таблице 7 будут храниться коды строк, соответствующие значениям (код_элемента – соответствующий продольному отверстию; код_шероховатости_Ra – соответствующий шероховатости 5; код_шероховатости_Rz – нет; код_квалитет – соответствующий квалитету 7).

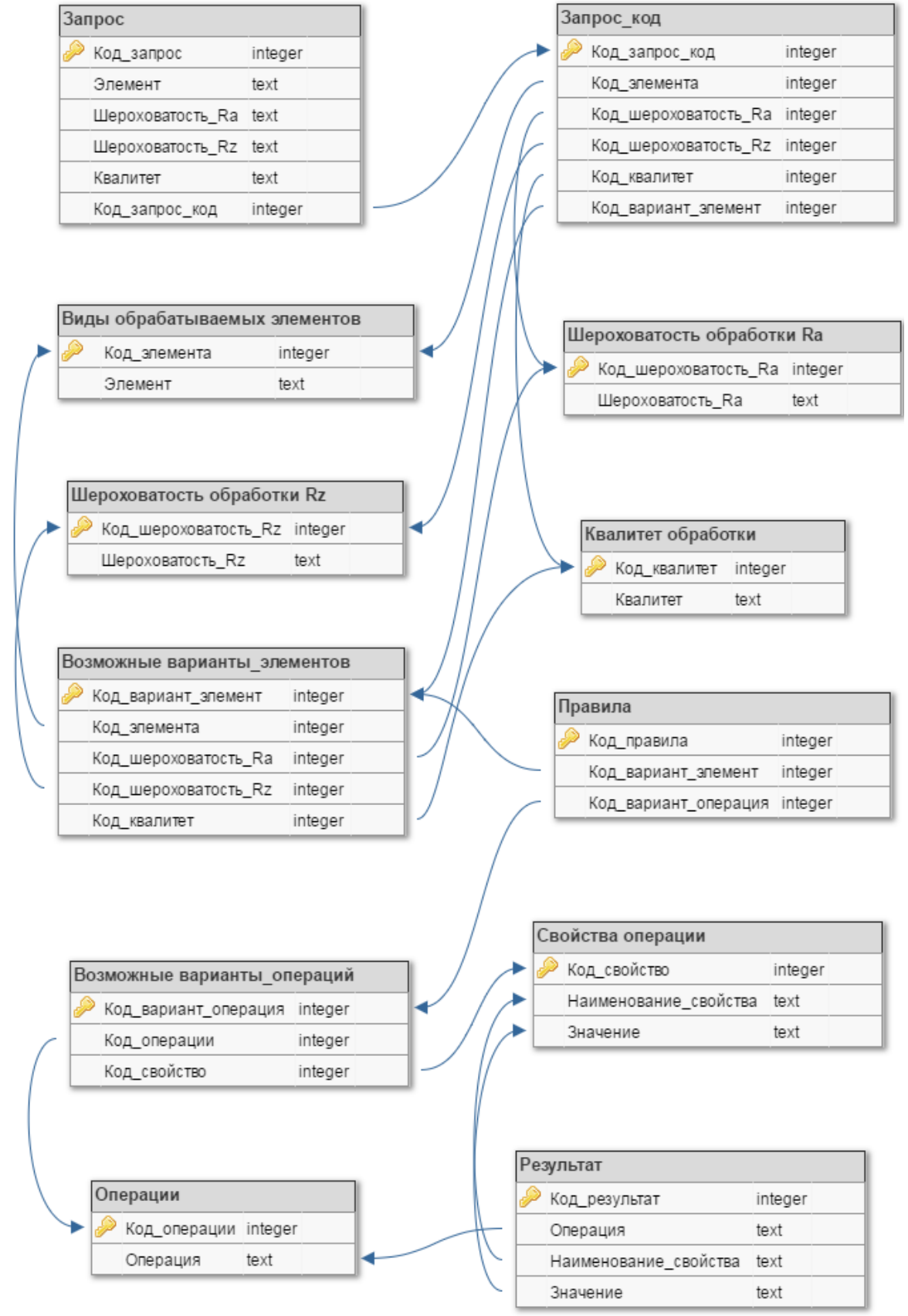


Рис. 1. Структура базы данных хранения онтологии
для составления технологии изготовления валов

В таблице 8 будут храниться коды строк, соответствующие значениям (код_операции – соответствующий операции сверление; код_свойства – соответствующий чистоте – чистовая операция) В таблице 9 хранится соответствие значений из таблицы 7 (код_элемента – соответствующий продольному отверстию; код_шероховатости_Ra – соответствующий шероховатости 5; код_шероховатости_Rz – нет; код_квалитет – соответствующий квалитету 7) и таблицы 8 (код_операции – соответствующий операции сверление; код_свойства – соответствующий чистоте – чистовая операция).

Далее обработчик записывает значения таблиц 5 и 6, соответствующие значениям ячейки «код» таблицы 8, полученной из таблицы 9.

В таблице 9 хранится соответствие значений из таблицы 7 (код_элемента – соответствующий продольному отверстию; код_шероховатости_Ra – соответствующий шероховатости 5; код_шероховатости_Rz – нет; код_квалитет – соответствующий квалитету 7) и таблицы 8 (код_операции – соответствующий операции сверление; код_свойства – соответствующий чистоте – чистовая операция).

Далее обработчик записывает значения таблиц 5 и 6, соответствующие значениям ячейки «код» таблицы 8, полученной из таблицы 9.

Таким образом осуществляется ассоциация определенного вида элемента с определенного вида операцией.

Представленная структура базы знаний для автоматизированного выбора технологии изготовления деталей типа «Вал» используется в Тамбовском государственном техническом университете при разработке системы автоматизированного проектирования технологического оборудования [1–8].

Библиографический список

1. Новые информационные технологии в открытом инженерном образовании / Е. Н. Малыгин, С. В. Карпушкин, М. Н. Краснянский, В. Г. Мокрозуб, А. Б. Борисенко // Новые информационные технологии в открытом инженерном образовании. – М. : Машиностроение-1, 2003. – 124 с.

2. Малыгин, Е. Н. Система автоматизированного расчета и конструирования химического оборудования / Е. Н. Малыгин, С. В. Карпушкин, В. Г. Мокрозуб, М. Н. Краснянский // Информационные технологии. – 2000. – № 12. – С. 19–21.

3. Разработка прототипа виртуальной модели учебно-материальных ресурсов университета химико-технологического профиля / В. А. Немтинов, В. В. Юханов, Е. Н. Малыгин, С. В. Карпушкин, С. Я. Егоров, В. Г. Мокрозуб, А. Б. Борисенко, Ю. В. Немтинова // Вопросы современной науки и практики. Университет им. В. И. Вернадского. – 2013. – № 3 (47). – С. 321–330.

4. Мокрозуб, В. Г. Структура интеллектуальной автоматизированной системы механических расчетов технологического оборудования / В. Г. Мокрозуб, А. В. Мокрозуб, А. А. Чуриков // Вопросы современной науки и практики. Университет им. В. И. Вернадского. – 2015. – № 4 (58). – С. 187–195.

5. Мокрозуб, В. Г. Автоматизированная информационная система подготовки производства машиностроительного предприятия / В. Г. Мокрозуб, А. Н. Поляков, А. И. В. Сердюк, К. В. Марусич, М. В. Овечкин // Вестник Тамбовского государственного технического университета. – 2012. – Т. 18, № 3. – С. 598–603.

6. Мокрозуб, В. Г. Структура информационно-логической модели кожухотрубчатых теплообменников / В. Г. Мокрозуб, С. В. Морозов // Вестник Тамбовского государственного технического университета. – 2013. – Т. 19, № 3. – С. 518–526.

7. Егоров, С. Я. Автоматизированная информационная система поддержки проектных решений по компоновке промышленных объектов. Чсть 2. Структура и функционирование системы (ч. 1 см. в ИТПП № 4, 2009 г.) / С. Я. Егоров, В. Г. Мокрозуб, В. А. Немтинов, М. С. Громов // Информационные технологии в проектировании и производстве. – 2010. – № 1. – С. 33–39.

8. Мокрозуб, В. Г. Разработка интеллектуальных информационных систем автоматизированного проектирования технологического оборудования : учеб. пособие / В. Г. Мокрозуб. – Тамбов : Тамбовский ГТУ, 2008. – 80 с.

В. А. Бочаров, А. А. Шивидова

Московский государственный технический университет
им. Н. Э. Баумана, Москва, Россия

РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ ДЛЯ СТРУКТУРИЗАЦИИ ТРЕБОВАНИЙ ДЛЯ ПРОЕКТИРОВАНИЯ КОНСТРУКТОРСКОЙ СТРУКТУРЫ ИЗДЕЛИЯ

Приведены актуальные причины разработки системы, которая отдельно от PLM-системы выполняет структуризацию технических требований на предприятии, а также даны требования для данной системы. Даны понятия требования в производстве. Объяснено, как идет процесс проектирования изделия на основе требований.

Введение

Проектирование наукоемкой продукции, на основании ГОСТ 2.103–68 [1], состоит из 5 этапов: «Техническое предложение», Аванпроект, Эскизный проект, Технический проект и Рабочий проект.

Для этапа «Техническое предложение» исходными данными являются первичные требования. Техническое предложение – совокупность конструкторских документов, которые должны содержать технико-экономические обоснования целесообразности разработки изделия на основании: 1) анализа технического задания заказчика и различных вариантов возможных конструктивных и технологических решений; 2) сравнительной оценки вариантов решений с учетом конструктивных и эксплуатационных особенностей разрабатываемого изделия и существующих аналогов и др.

На этапе аванпроекта отдел главного конструктора и службы маркетинга преобразуют требования заказчика в структуру требований для конструкторов, составляют ТЗ и создают первичную компоновку изделия, уточняя детали у заказчика.

На этапе эскизного проектирования формируется календарный план работ, выполняются инженерные расчеты для выбранной компоновки изделия, готовится электронный макет верхнего уровня (компоновочный).

На этапе технического проектирования разрабатывается детализированный электронный макет изделия, выполняются необходимые виды инженерного анализа предложенного конструкторского решения, производится оценка его технологичности, формируются списки используемых стандартных и покупных изделий. В ходе технического проектирования осуществляется валидация [2] требований, закрепленных в техническом задании.

Требования появляются на первом этапе и связаны с изделием на всех этапах. Само понятие требования очень широкое, используем определение, которое чаще применяется, в соответствии с ГОСТ 2.114–95 [3], в машиностроении: требование – это утверждение, которое идентифицирует целевые эксплуатационные, функциональные параметры, характеристики или ограничения проектирования продукта или процесса, которое однозначно проверяемо и измеримо.

Актуальность задачи информационной поддержки управления требованиями заключается в том, что от качества процессов управления требованиями при проектировании технически сложной продукции в конечном счете зависят трудоемкость и эффективность реализации эскизного и технического проекта, техническая сложность и потенциал развития самого проекта. Для конструкторского бюро формирование дерева требований и связь его с конструкторским составом разрабатываемого изделия является первоочередной задачей после уточнения ТЗ.

Функции информационной поддержки процессов управлениями проектными требованиями реализуются в ведущих PDM-системах, позиционируемых как основа PLM-решения на предприятии. К числу таких систем можно отнести Teamcenter (Siemens) и

SmarTeam (Dassault Systemes). Однако приобретение пакета лицензий и организация комплекса работ по широкомасштабному внедрению указанных систем на предприятии является крайне дорогостоящим мероприятием. В то же время у представителей промышленности возникает интерес к программным инструментам поддержки проектных процедур на ранних стадиях проектирования (в том числе – с учетом решений, полученных в ранее выполненных проектах).

Процесс проектирования КСИ на основе требований

В предыдущем пункте было дано понятие, что такое, а теперь осталось понять, как же происходит проектирование КСИ под управлением требований.

Сначала из ТЗ заказчика узнается, какое изделие в конечном итоге должно получится. Потом уже формируется скелет дерева требований, который описывает основные характеристики изделия. Требования могут быть вариативными, то есть ТЗ заказчика можно описать не одним деревом требований, а несколькими, что позволяет найти оптимальный вариант для решения задач. Позже составляется основа функциональной структуры, которая описывает главные функции и показывает, каким параметрам должны они соответствовать. На основе этих базовых структур формируется скелет КСИ, где будут все основные элементы, которые реализуют данные функции. В последствии, когда структура требований и ФСИ детализируются, расширяется КСИ до уровня деталей. Последним этапом являются детали/подсборки, связанные уже с конкретными проверками требований на соответствие их выполнения. Так же существует вариативность проверок требований, т.е. одно требование может быть проверено разными способами, это приводит к тому, что есть возможность сравнить результаты проверок, полученные разными способами и понять, какой метод проверки надежней и точней.

Создание вариантов существует и в проектировании ФСИ, тут уже появляется понятие – принятие решений. Принятие решений – когнитивный процесс, результатом которого является выбор мнения или курса действий среди нескольких альтернативных возможностей. Например, есть функция охлаждения прибора, можно реализовать эту функцию с помощью вентиляционной системы, а можно с помощью системы жидкостного охлаждения (СЖО). Второй вариант более дорогой и более сложный, но гораздо эффективнее справляется с поставленной задачей. Здесь как раз конструктор должен принять решение, что использовать воздушное охлаждение или СЖО.

Разработка требований к информационной системе и прототип

Актуальность разработки информационной системы, в которой можно создавать СТ, ФСИ и КСИ, и делать трассировку, заключается в нескольких факторах:

- многие КБ создают требования на бумаге и от сюда возникает проблема передачи требований по заводу и увеличения риска ошибки при формировании самой структуры требования;
- PLM-система Teamcenter очень дорогая и не каждое предприятие может себе его позволить;
- и нужно, чтобы при просмотре требований на компьютере была видна иерархия всех 3-х структур, и как они связаны между собой.

Перед тем как начать разрабатывать такую систему, нужно сформулировать требования к ней, какими сущностями она обладает. В системе будут присутствовать 8 сущностей:

1) спецификация требования (в иерархии структуры требований занимает самый верхний уровень, от нее могут создавать под спецификации, требования);

2) под спецификации (разбивают большую задачу на маленькие, для простоты восприятия);

3) требования (по иерархии идут ниже спецификации и в них могут создавать проверки);

4) проверки требований (самый низкий уровень, в них описывается, что проверяем и по каким критериям);

5) функции (верхний уровень ФСИ, в ней создаются под функции);

6) под функции (в них описывается часть логики всей функции);
7) сборки и детали;
8) трассировки.

Система принимает данные из БД и так же сохранять в нее все изменения. Сущности можно создавать, корректировать, удалять. Следующим требованием к системе будет отображение матрицы трассируемости и трассировки. Система должна отображать матрицу трассировки и для всех требований и для функций, т.е. можно выбрать показать связи для ФСИ или для СТ, а также должна подсчитывать сколько всего связей у каждого элемента.

После формулировки требований переходим к реализации прототипа. Сначала была выбрана БД MS SQL 2016, как простое и удобное хранилище данных для разработчика и для ОС. Разработка ведется в Windows Server 2016 с помощью технологии MVC 5 и языков C#, HTML, Java-Script, а также с помощью CSS был создан дизайн. Дальше были рассмотрены варианты создания иерархии в БД:

1. Список смежности.
2. Материализованный путь.
3. Вложенные множества.

Рассмотрев все эти варианты, был выбран метод «Список смежных вершин», так как он более удобен для редактирования и добавления новых узлов, в произвольные места дерева. Более подробное описание этих методов и сравнение между собой есть в источнике [4].

Но есть еще так называемые графовые БД. Это разновидность баз данных с реализацией сетевой модели в виде графа и его обобщений. Такой принцип работы с данными будет удобен, если в данной системе добавить сравнение данных, так как по графу искать данные удобнее, то результаты команд будут лучше. Это нужно, если сделать автоматическую генерацию требований. Допустим в БД уже есть проекты, в которых есть требования, и нужно создать новый проект, похожий на один из них. Тогда пусть система найдет по определенным критериям те данные, которые максимально близки для новой структуры требований, и сама сгенерирует новые требования. Такой подход резко уменьшит затраты на то, чтобы самому с нуля все делать, так как скелет будет уже готов, его нужно только дополнить и подправить, если требуется. Так же можно сделать и для ФСИ и КСИ. Такая система позволяет быстрее выпускать продукцию и дает больше времени на модернизацию изделий. Графовая БД лучше работает на больших объемах информации и поиск работает в несколько раз быстрее, чем в обычных БД. Было проведено сравнение между MySQL и Neo4j, первая База Данных, ядро которой основано на теории графов и выведены графики, один из которых изображен на рис. 1.

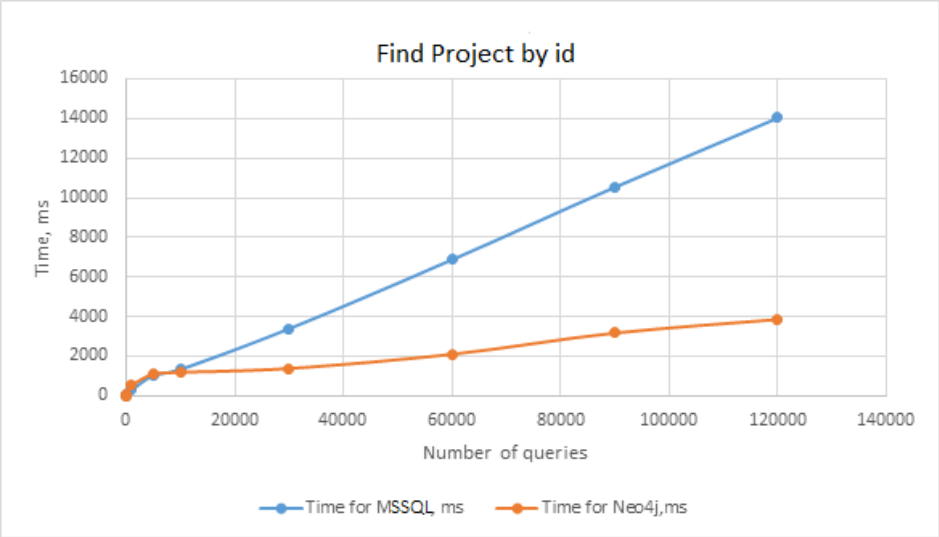


Рис. 1. Результаты эксперимента

Для более четкого представления разницы между этими БД можно понять по модели Графовой Базе Данных (рис. 2).

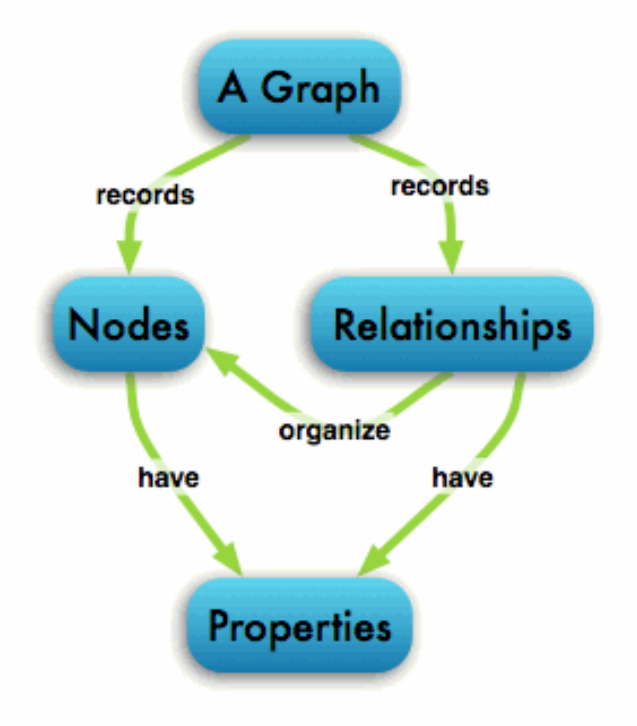


Рис. 2. Схема модели в Графовой БД

Реализация разработки информационной системы ведется по CASE-технологии. CASE (computer-aided software engineering) – набор инструментов и методов программной инженерии для проектирования программного обеспечения, который помогает обеспечить высокое качество программ, отсутствие ошибок и простоту в обслуживании программных продуктов https://ru.wikipedia.org/wiki/CASE – cite_note-1 Также под CASE понимают совокупность методов и средств проектирования информационных систем с использованием CASE-инструментов. Основной целью CASE-технологии является разграничение процесса проектирования программных продуктов от процесса кодирования и последующих этапов разработки, максимально автоматизировать процесс разработки. Для выполнения поставленной цели CASE-технологии используют два принципиально разных подхода к проектированию: структурный и объектно-ориентированный.

Вывод

В итоге данная система сделает большой прорыв в производстве на начальных этапах всех проектов и дает новую планку для систем, позволяющие управлять жизненным циклом изделия, что позволяет увеличить конкурентно-способность предприятия и уменьшает количество ошибок при формировании изделий.

Библиографический список

1. ГОСТ 2.103–68. Стадии проектирования. – М. : Изд-во стандартов, 1970. – 3 с.
2. Валидация – Википедия. – URL: https://ru.wikipedia.org/wiki/Валидация (дата обращения: 1.05.2017).
3. ГОСТ 2.114–95. Технические условия. – М. : Изд-во стандартов, 1995. – 12 с.
4. Иерархические структуры данных и Doctrine. Сайт Хабрахабр. – URL: https://habrahabr.ru/post/46659/ (дата обращения: 10.05.2017).

А. В. Вилюха

Брянский государственный технический университет, Россия, Брянск

АВТОМАТИЗАЦИЯ КОНТРОЛЯ КАЧЕСТВА ШТАМПОВАННЫХ КОНТАКТОВ С ПОМОЩЬЮ МЕТОДА СЖАТИЯ-РАСШИРЕНИЯ

Рассмотрена проблема контроля качества штампованных электрических контактов. Для разрешения данной проблемы предлагается использовать метод автоматизированного контроля качества. Рассмотрена схема реализации метода сжатия-расширения для контроля внешнего контура и отверстий контактов.

Серийное производство прямоугольных электрических соединителей требует инженерных решений высокого качества не только на стадии проектирования, но также на стадии производства и контроля качества выпускаемых изделий. Одним из ключевых технологических процессов является штамповка электрических контактов. Высокая повторяемость геометрических характеристик штампованных изделий позволяет применять к ним периодический контроль качества. Однако данный вид контроля качества не позволяет своевременно выявить бракованные изделия, что ведет к увеличению затрат на производство партии контактов и в конечном счете негативно сказывается на конечной стоимости электрического соединителя.

Решить данную проблему можно при помощи использования автоматизированных систем контроля качества. Данные системы работают в режиме реального времени с изображением получаем с камеры, установленной над лентой с контактами, и позволяют остановить процесс штамповки при выявлении первого бракованного контакта.

Для обработки изображений используются различные алгоритмы. Одним из наиболее простых в реализации является метод направленного сжатия и растяжения, который позволяет быстро и эффективно контролировать внешний контур и отверстия контакта.

Метод направленного сжатия и расширения не требует изображение-эталон для проведения анализа, использует универсальный математико-логический аппарат.

Суть метода заключается в следующем. Пусть каждый элемент a_{ij} матрицы, представляющий контролируемую топологию, имеет четыре соседних элемента: два из них $a_{i-1,j}$, $a_{i+1,j}$ − соседние в строке и два $a_{i,j-1}$, $a_{i,j+1}$− соседние в столбце. Операцией расширения (сжатия) на один разряд принимаем операцию преобразования изображения, заключающуюся в том, что каждому элементу a_{ij} присваивается единичное (нулевое) значение, если хотя бы один из четырех его соседних элементов имеет единичное (нулевое) значение; остальные элементы свои значения не изменяют. Операции расширения и сжатия на один разряд описываются формулами соответственно:

$$a_{ij} = a_{i-1,j} \cup a_{i+1,j} \cup a_{i,j-1} \cup a_{i,j+1}; \quad (1)$$

$$a_{ij} = a_{i-1,j} \cap a_{i+1,j} \cap a_{i,j-1} \cap a_{i,j+1}. \quad (2)$$

Обобщенная операция расширения (сжатия) на k разрядов заключается в последовательном применении к изображению k операций расширения (сжатия) на один разряд.

Последовательное применение сжатия, а потом расширения позволит «устранить» наплывы и пятна, заполняющие расстояние между конструктивными элементами контакта. Таким образом, применение операций расширения и сжатия позволяет восстановить эталон.

Недостатком метода является трудность локализации крупномасштабных дефектов, т.е. дефектов, размеры которых сравнимы или превышают минимальную ширину контакта [1].

На начальных этапах разработки программы возникли следующие противоречия, между исходными данными и принципами работы метода:

• изображение с камер, поступающее для анализа является цветным, а метод работает с черно-белым изображением;

• реализуемый алгоритм восстанавливает эталон, но не локализует область расположения дефектов, которые возможно устранить.

Решить данные противоречия, возможно добавив этапы переведения цветного изображения в черно-белое и сравнения восстановленного эталона с исходным изображением с последующим отображением на экране областей несоответствия. Схема реализации представлена на рис. 1.

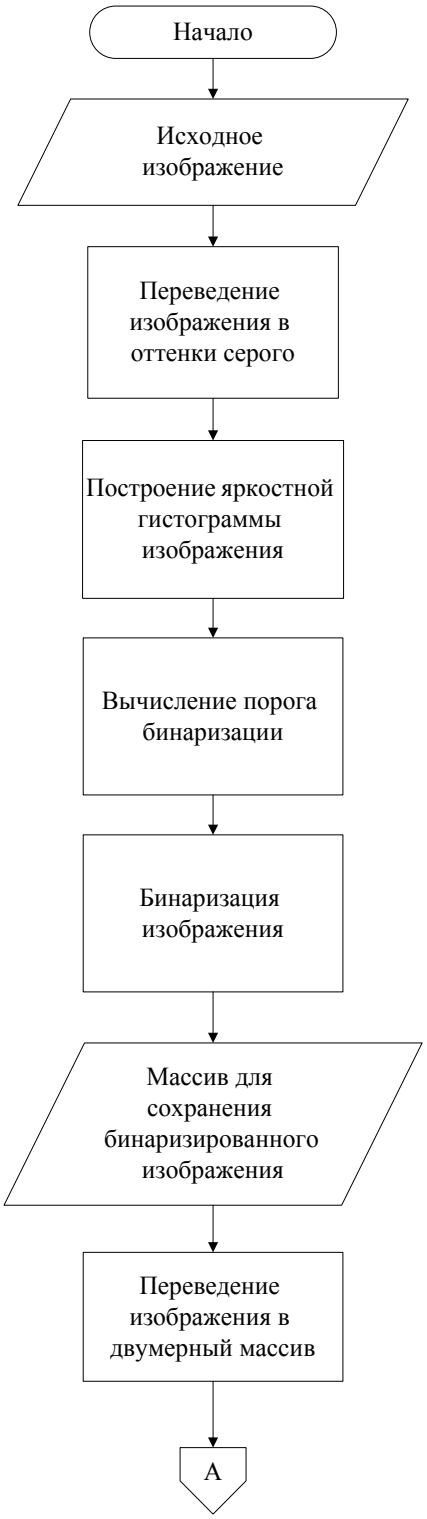


Рис. 1. Схема реализации метода (начало)

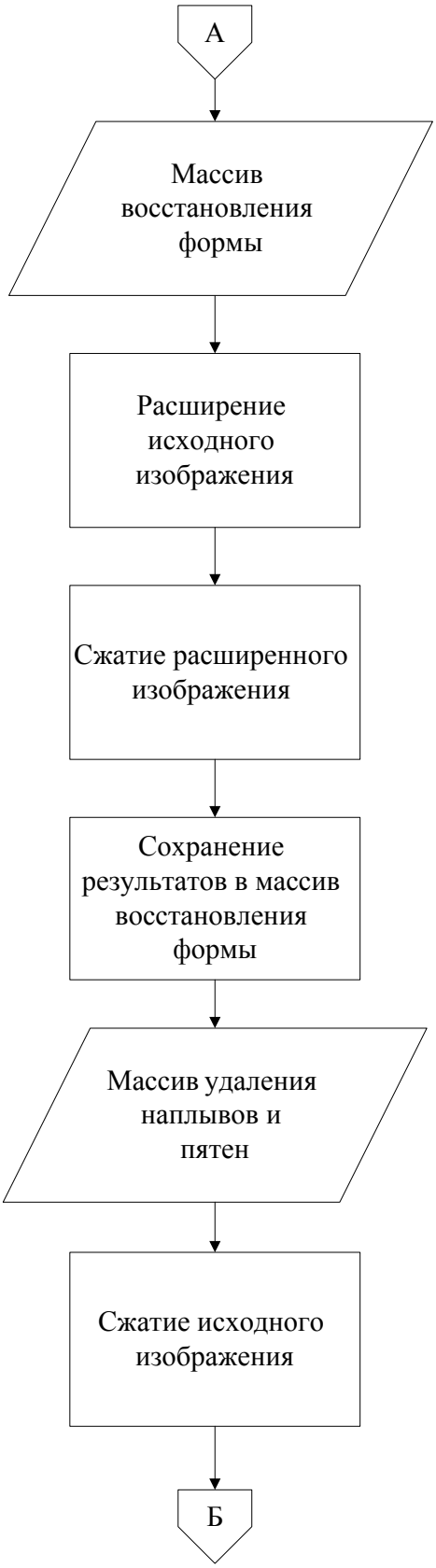


Рис. 1. Схема реализации метода (продолжение)

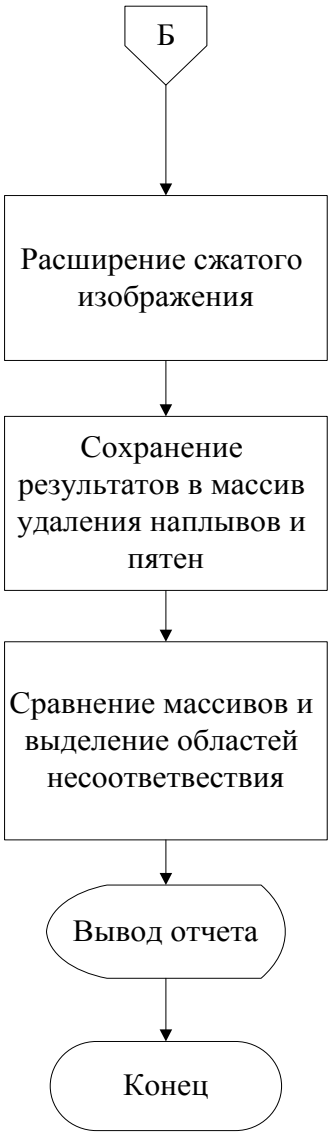


Рис. 1. Схема реализации метода (окончание)

Использование автоматизированных систем контроля качества для штампованных электрических контактов позволит снизить затраты на производство электрических соединителей. В качестве методики проверки внешнего контура и отверстий штампованных контактов можно использовать метод направленного сжатия-расширения, для которого разработана схема реализации.

Библиографический список

1. Бутаков, Е. А. Обработка изображений на ЭВМ / Е. А. Бутаков, В. И. Островский, И. Л. Фадеев. – М. : Радио и связь, 1987. – 240 с.

2. Сафонов, Л. И. Прямоугольные электрические соединители. Основные принципы системы менеджмента качества предприятия, выпускающего радиоэлектронные компоненты / Л. И. Сафонов, А. Л. Сафонов // Технологии в электронной промышленности. – 2008. – № 1. – С. 68–71.

3. Смирнов, Н. А. Современные методы анализа и контроля продуктов производства / Н. А. Смирнов. – М. : Металлургия, 1985.

4. Braunovic, M. A model for lifetime evolution of closed electrical contacts / M. Braunovic, V. V. Izmailov, M. V. Novoselova // Proceeding of 51th JEEE Holm Conference on Electrical Contacts. – Chicago, 2005. – P. 217–223.

Н. С. Доброва

Тамбовский государственный технический университет, Тамбов, Россия

АВТОМАТИЗИРОВАННЫЙ ВЫБОР ТЕХНОЛОГИЧЕСКИХ ПРОЦЕССОВ ТЕРМИЧЕСКОЙ ОБРАБОТКИ ИЗДЕЛИЙ ИЗ МЕТАЛЛОВ[1]

Рассмотрены вопросы автоматизации процесса проектирования технологических процессов химико-термической обработки изделий из металлов, в частности, поставлена задача автоматизированного выбора экономичного экологически безопасного технологического процесса, оборудования, приспособлений, вспомогательных материалов и режимных параметров определенного ранее вида упрочняющей обработки, а также предложен алгоритм ее решения.

Для высокого качества выпускаемой продукции предприятию необходимо использовать детали хорошего качества. Для эффективной и долговечной работы приборов при сборке необходимо использовать элементы хорошего качества, прошедшие качественную и технологически правильную обработку. Особыми элементами изделия являются элементы выполненные из металла [1–2].

Требуемые физико-механические свойства металлических деталей достигаются с помощью термической обработки. Термообработка позволяет работать над свойствами металлов без изменения их химического состава. Основа данного процесса состоит в нагревании металла до нужной определенному виду материала температуры. За тем происходит выдержка металла в жидкой среде до достижения материала нужных свойств. В зависимости от выбранного режима нагрева, вида термической обработки и температуры охлаждения можно получить абсолютно уникальную структуру и свойства материала.

Для достижения желаемых свойств металлов на предприятиях создаются участки термической обработки металлических изделий. Существует несколько методик проведения термообработки. Основными видами термической обработки являются: отжиг; закалка; нормализация; отпуск; криогенная обработка.

Например, для того чтобы получить равновесную структуру стали, используют такой вид термообработки металлов, как отжиг 1-го рода. Сюда можно отнести такие подвиды термообработки металлов, как гомогенизация, рекристаллизация и снятие напряжений. Этот отжиг не влияет на внешние превращения, а лишь делает структуру металла равномерной и правильной. Отжиг 2-го рода способен влиять на внешние превращения, к нему можно отнести нормализацию, полный и неполный отжиг. С помощью данного способа можно упрочнить металл в несколько раз.

Существует так же несколько разновидностей печей, Шахтные печи для термообработки различных размеров. Используют для нескольких видов термообработки: для нагрева под закалку, для отжига, отпуска, цементации. Подходят для термообработки цветных сплавов, где технологией не предусмотрена высокая точность технологических параметров и скорость переноса садки из печи в закалочную среду.

Камерные печи для термической обработки применяются в большинстве технологий для термической обработки средних и мелких деталей. Подходят для любых типов производств и для любых технологий обработки. Их можно используют и в составе гибких автоматизированных комплексов, и как отдельно стоящие единицы оборудования.

Для термообработки крупногабаритных и массивных деталей и узлов применяют печи с выдвижным подом. При загрузки и выгрузки обычно применяют краны и кранбалки. К недостаткам данных видов печей относят большие теплопотери и большие габариты за счет выдвижного пода.

[1] Работа выполнена под руководством заведующего кафедрой КИСМ ТГТУ В. А. Немтинова.

Отличный от традиционного способа нагрева является метод индукционного нагрева ТВЧ. Нагреваемая деталь помещается в электромагнитное высокочастотное поле индуктора (согнутая по конфигурации детали медная трубка). При этом в детали индуктируются переменные, так называемые, токи высокой частоты, плотность которых неравномерна по сечению, они оттесняются возникающим внутри детали быстропеременным магнитным полем к поверхности нагреваемой детали (так называемый поверхностный эффект).

Индукционная или высокочастотная термообработка получила широкое распространение во всех отраслях машиностроения и имеет самое разнообразное назначение: для закалки деталей и инструмента, нормализации, пайки, отпуска, плавки, нагрева под ковку и штамповку, напайки пластин из твердого сплава.

Контроль качества играет важную роль в технологических процессах термической обработки. Для постоянного контроля на производстве используются стационарные твердомеры Роквелл и Бринелль. Для определения соответствия качества измерения проводятся либо непосредственно на деталях, либо контрольных образцах. Для крупногабаритных изделий используются портативные твердомеры с прямым методом измерения и приборы для косвенного измерения механических свойств. Приборы такой разновидности измеряют различные физические величины, которые напрямую зависят от твердости, прочности, пластичности или вязкости. Контроль химико-термической обработки производят как по твердости, так и по глубине слоя на образцах-свидетелях при помощи портативного микроскопа, с нанесенной на объектив линейкой. Таким образом, в цехах термообработки используется просто огромное количество основного и вспомогательного оборудования, основная цель использования которого – обеспечение требуемых свойств металлических изделий [3–8].

В связи с этим в данной работе рассматриваются вопросы автоматизации процесса проектирования технологических процессов химико-термической обработки изделий из металлов, в частности: поставлена задача автоматизированного выбора экономичного экологически безопасного технологического процесса, оборудования, приспособлений, вспомогательных материалов и режимных параметров определенного ранее вида упрочняющей обработки, а также предложен алгоритм ее решения. Апробация методики решения задачи выполнена на ряде деталей, выпускаемых на ОАО «Завод Тамбовполимермаш».

Библиографический список

1. Инфомационные и процедурные модели синтеза экологически безопасных технологических процессов химико-термической обработки изделий из металлов : учеб. пособие / Е. Н. Малыгин, В. А. Немтинов, В. Г. Егоров, В. Г. Мокрозуб, В. Г. Однолько. – Тамбов : Изд-во Тамб. гос. техн. ун-та, 2007. – 108 с.

2. Оборудование для термической обработки: основные виды печей. – URL: https:// heattreatment.ru/oborudovanie-dlya-termicheskoj-obrabotki-osnovnye-vidy-pechej.html (дата обращения: 05.10.2017).

3. Разработка прототипа виртуальной модели учебно-материальных ресурсов университета химико-технологического профиля / В. А. Немтинов, В. В. Юханов, Е. Н. Малыгин, С. В. Карпушкин, С. Я. Егоров, В. Г. Мокрозуб, А. Б. Борисенко, Ю. В. Немтинова // Вопросы современной науки и практики. Университет им. В. И. Вернадского. – 2013. – № 3 (47). – С. 321–330.

4. Мокрозуб, В. Г. Применение N-ориентированных гиперграфов и реляционных баз данных для структурного и параметрического синтеза технических систем / В. Г. Мокрозуб, В. А. Немтинов, А. С. Мордвин, А. А. Илясов // Прикладная информатика. – 2010. – № 4 (28). – С. 115–122.

5. Немтинов, В. А. Применение теории нечетких множеств и экспертных систем при автоматизированном выборе элемента технической системы / В. А. Немтинов, С. Я. Егоров, П. И. Пахомов // Информационные технологии. – 2009. – № 10. – С. 34–38.

6. Немтинов, В. А. Информационная модель объекта сложной технической системы / В. А. Немтинов, В. Г. Мокрозуб, Е. С. Егоров // Радиотехника. – 2010. – № 12. – С. 21–23.

7. Немтинов, В. А. Оперативное управление выпуском продукции с использованием системы моделирования динамических процессов / В. А. Немтинов, Ю. В. Немтинова, Д. С. Русских // Вестник Тамбовского государственного технического университета. – 2007. – № 2. – С. 372–378.

8. Зимнухова, Ж. Е. О подходе к построению автоматизированной информационной системы поддержки принятия решений для проектирования процессов производства изделий из металлов / Ж. Е. Зимнухова, В. А. Немтинов // Информационные технологии. – 2008. – № 9. – С. 29–34.

М. С. Елагин, В. А. Мокрозуб, Н. В. Храмцова

Тамбовский государственный технический университет, Тамбов, Россия

ЗАДАЧА МИНИМИЗАЦИИ МЕТАЛЛОЕМКОСТИ ВЕРТИКАЛЬНЫХ АППАРАТОВ[1]

Для заданного объема и давления находится диаметр, при котором будет минимальный вес аппарата.

В обрабатывающих отраслях промышленности широко применяются емкостные аппараты для переработки и хранения сырья и готовой продукции. Основным элементом этих аппаратов является корпус, состоящий из обечайки, днища и крышки. Корпус в значительной степени определяет металлоемкость и стоимость аппарата. Толщина стенок элементов корпуса зависит от рабочего давления и температуры в аппарате.

Расчет толщины корпуса осуществляется по ГОСТ Р 52857.2–2007 «Сосуды и аппараты. Нормы и методы расчета на прочность. Расчет цилиндрических и конических обечаек выпуклых и плоских днищ и крышек». В соответствии с этим стандартом толщина цилиндрической обечайки, нагруженной внутренним давлением, зависит от диаметра аппарата, давления, допускаемого напряжения материала и компенсационной прибавки, которая учитывает коррозию материала, минусовой допуск листов и утонение стенок при изготовлении обечайки.

В настоящее время на кафедре «Компьютерно–интегрированные системы в машиностроении» разрабатывается система автоматизированного конструирования технологического оборудования [1–5]. Программы различных расчетов выполненные в среде Mathcad являются элементами этой системы [6–11].

Задача расчета оптимальных размеров аппарата ставится следующим образом.

Даны тип аппарата, объем, рабочее давление, температура и материал изготовления корпуса аппарата. Требуется найти такой диаметр аппарата, D^*, при котором металлоемкость корпуса, M_k будет минимальна.

$$M_k(D^*) = \min_{D\min<D<D_{\max}} M_k(D) \ ,$$

при следующих ограничениях (аппарат с эллиптическим днищем и эллиптической крышкой):

$$M_k = \rho\left(A_o S_o + A_к S_к + A_д S_д\right);$$

$$S_o = PD/(2\varphi[\sigma] - P) + c;$$

$$S_д = S_k = PD/(2\varphi[\sigma] - 2P) + c;$$

$$A_o = \pi DL;$$

$$A_к = A_д = 1,24D^2;$$

$$V = L\pi D^2/4 + \pi D^3/24 + \pi D^3/24$$

Обозначения:

$D\min, D\max$ – минимальный и максимальный допустимый диаметр аппарата;

[1] Работа выполнена под руководством профессора кафедры КИСМ ТГТУ В. Г. Мокрозуба.

P – внутреннее давление в аппарате;

ρ – плотность материала корпуса аппарата;

$A_о$, $A_к$, $A_д$ – площадь поверхности обечайки, днища и крышки соответсвенно;

$S_о$, $S_к$, $S_д$, – толщина обечайки, крышки и днища соответственно

L – высота обечайки;

V – объем аппарата;

$[\sigma]$ – допускаемое напряжение материала аппарата;

φ – коэффициент прочности сварных швов.

Библиографический список

1. Мокрозуб, В. Г. Представление модели параметрического синтеза технического объекта в реляционной базе данных / В. Г. Мокрозуб, А. И. Сердюк, С. Ю. Шамаев, С. В. Каменев // Вестник Тамбовского государственного технического университета. – 2011. – Т. 17, № 2. – С. 462–466.

2. Мокрозуб, В. Г. Представление структуры технических объектов с взаимозаменяемыми элементами в виртуальных моделях / В. Г. Мокрозуб, А. И. Сердюк, С. В. Каменев, С. Ю. Шамаев // Вестник Тамбовского государственного технического университета. – 2011. – Т. 17, № 2. – С. 467–470.

3. Мокрозуб, В. Г. Создание виртуального кабинета «Конструирование технологического оборудования» в Тамбовском государственном техническом университете / В. Мокрозуб // САПР и графика. – 2015. – № 1 (219). – С. 38–39.

4. Егоров, С. Я. Автоматизированная информационная система поддержки проектных решений по компоновке промышленных объектов часть 1. Аналитические и процедурные модели (часть 2 см. 2010 № 1) / С. Я. Егоров, В. Г. Мокрозуб, В. А. Немтинов, И. В. Милованов // Информационные технологии в проектировании и производстве. – 2009. – № 4. – С. 3–11.

5. Егоров, С. Я Автоматизированная информационная система поддержки проектных решений по компоновке промышленных объектов часть 2. Структура и функционирование системы (часть 1 см. в ИТПП № 4, 2009 г.) / С. Я. Егоров, В. Г. Мокрозуб, В. А. Немтинов, М. С. Громов // Информационные технологии в проектировании и производстве. – 2010. – № 1. – С. 33–39.

6. Автоматизированная информационная система подготовки производства машиностроительного предприятия / В. Г. Мокрозуб, А. Н. Поляков, А. И. Сердюк, К. В. Марусич, М. В. Овечкин // Вестник Тамбовского государственного технического университета. – 2012. – Т. 18, № 3. – С. 598–603.

7. Мокрозуб, В. Г. Структура информационно-логической модели кожухотрубчатых теплообменников / В. Г. Мокрозуб, С. В. Морозов // Вестник Тамбовского государственного технического университета. – 2013. – Т. 19, № 3. – С. 518–526.

8. Мокрозуб, В. Г. Программное обеспечение автоматизированных систем размещения объектов в пространстве, инвариантное к предметной области / В. Г. Мокрозуб, К. В. Немтинов, К. А. Шаронин // Научно-техническая информация. Сер. 2, Информационные процессы и системы. – 2012. – № 3. – С. 11–20.

9. Мокрозуб, В. Г. Функциональная и процедурная модели проектирования технологического оборудования / В. Г. Мокрозуб // Программные системы и вычислительные методы. – 2014. – № 4. – С. 418–430. DOI: 10.7256/2305-6061.2014.4.13971.

10. Мокрозуб, В .Г Информационно-логические модели технических объектов и их представление в информационных системах / В. Г. Мокрозуб, В. А. Немтинов, С. Я. Егоров // Информационные технологии в проектировании и производстве. – 2010. – № 3. – С. 68–73.

11. Мокрозуб, В. Г. Интеллектуализация механических расчетов в виртуальном кабинете «Конструирование технологического оборудования» / В. Г. Мокрозуб // Промышленные АСУ и контроллеры. – 2014. – № 12. – С. 34–40.

Е. В. Мананкова

Тамбовский государственный технический университет, Тамбов, Россия

ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ РАБОТЫ ЭЛЕВАТОРНОГО КОМПЛЕКСА[1]

Представлен комплекс задач, решаемых с помощью построения модели бизнес-процессов предприятия, проанализированы самые распространенные нотации бизнес-моделирования, выявлены их преимущества, недостатки и область целесообразного применения.

Многие крупные компании отличаются отсутствием гибкости, обилием нерелевантной информации и отсутствием четкой формулировки задач. Данные барьеры препятствуют пониманию происходящих на предприятии (или проектируемых) процессов их участниками (проектировщиками). Одним из способов решения данной проблемы является построение бизнес-моделей.

Описание бизнес-процессов решает ряд задач, таких как:

– выявление недостатков и проблем технологий производства, организационной структуры, документооборота и автоматизации на этапе проектирования;

– распределение полномочий и ответственности за выполняемые работы среди сотрудников организации;

– проведение расчета оптимального количества специалистов для выполнения работ;

– формулировка требований к компетенциям и личным качествам сотрудников;

– выявление узких мест в производстве;

– оценка целесообразности автоматизации процесса;

– проведение анализа и оптимизации бизнес-процессов по времени и стоимости выполнения;

– ведение контроля деятельности предприятия по ключевым показа-телям.

Целью данной работы является построение модели бизнес-процессов элеватора, находящегося на этапе проектирования. Для выполнения данной работы были поставлены задачи: описать движение материальных потоков; описать документооборот; составить подробное описание процессов движения материальных потоков.

На данный момент активно используется большое количество методологий моделирования бизнес-процессов. Самыми популярными являются нотации IDEF, BPMN, EPC, DFD и блок-схемы.

Существует множество программных продуктов, позволяющих визуализировать процессы в соответствии с различными нотациями. В данной работе были использованы Microsoft Visio и ARIS express.

Microsoft Visio является современным многофункциональным решением для создания диаграмм и схем, которое позволяет упростить и связать информацию, а также поделиться ей. Данный программный продукт предоставляет множество графических возможностей и опций для создания собственных методов организации информации [1].

Visio рекомендован для ИТ-специалистов, разработчиков и аналитиков (например, связанных с бизнес-процессами, кадрами и управлением), которым требуется интерпретировать, обновлять и передавать сложную информацию о процессах, инфраструктуре и приложениях. Кроме того, диаграммы Visio можно связывать с несколькими источниками данных для детального отображения информации в реальном времени [2].

ARIS express – это инструмент, который разработан для моделирования бизнес-процессов. Пользователи отмечают простоту в установке и использовании данного ПО. ARIS express принадлежит к семейству средств моделирования ARIS (ARchitecture of Integrated Information Systems) компании IDS Scheer (являющейся частью фирмы Software

[1] Работа выполнена под руководством заведующего кафедрой КИСМ ТГТУ В. А. Немтинова.

AG), широко представленных на российском рынке. Семейство продуктов ARIS включает не только инструменты моделирования бизнес-процессов и публикации моделей, но и интегрирующиеся между собой средства разработки системы сбалансированных показателей, оценки и оптимизации стоимости бизнес-процессов, их имитационного моде-лирования, инструменты, упрощающие внедрение ERP-систем, проектирование распределенных приложений и ИТ-инфраструктуры, а также инструменты контроля за выполнением бизнес-процессов. Ведущие аналитические компании Gartner Group и Forrester Research относят компанию IDS Scheer к лидерам мирового рынка средств моделирования и анализа бизнес-процессов [2]. К минусам данного продукта можно отнести отсутствие русскоязычной версии и невозможность применения дополнительных элементов, не входящих в используемую нотацию.

В ходе разработки бизнес-модели функционирования проектируемого элеватора был проведен анализ технологической карты, согласно которой можно проследить движение материальных потоков.

Изначально была разработана модель движения потоков зерна посредством программного продукта Microsoft Visio. На данной схеме отражена последовательность процессов, происходящих в различных узлах элеватора. Для более точного отображения и информативности были объединены нотация EPC и блок-схемы.

Выполнение данной работы привело к решению следующих задач: определение необходимых маршрутов движения зерна; учет зерна, проходящего процессы очистки и сушки в элеваторе; отображение документооборота, связи с внешними процессами; детальное описание процессов в нотации BPMN 2.0.

Затем каждый процесс был подробно описан в нотации BPMN 2.0 в ARIS express. Данные модели подробно описывают функции оборудования и позволяют соотнести процессы с позициями технологической схемы. Они необходимы для программирования автоматических систем управления и составления должностных инструкций персонала [3–9].

Таким образом, в результате работы были решены поставленные выше задачи. Использование предложенной модели при проектировании элеваторного комплекса позволило создать представление процессов у всех участников проекта, что позволило сократить временные ресурсы. Также, в результате анализа технологической схемы были найдены возможные маршруты движения зерна внутри элеватора, которые противоречат логике происходящих процессов и концепции учета сырья.

Библиографический список

1. URL: http://www.microsoft.com/ru-ru/office/vip/visio.aspx
2. URL: http://www.ariscommunity.com/aris-express
3. Разработка прототипа виртуальной модели учебно-материальных ресурсов университета химико-технологического профиля / В. А. Немтинов, В. В. Юханов, Е. Н. Малыгин, С. В. Карпушкин, С. Я. Егоров, В. Г. Мокрозуб, А. Б. Борисенко, Ю. В. Немтинова // Вопросы современной науки и практики. Университет им. В. И. Вернадского. – 2013. – № 3 (47). – С. 321–330.
4. Имитационное моделирование динамических процессов при управлении городским пассажирским транспортом / В. А. Немтинов, Н. В. Пеньшин, Ю. А. Донских, К. В. Немтинов, Е. С. Егоров // Информационные технологии. – 2012. – № 4. – С. 75.
5. Немтинов, В. А. Информационная модель объекта сложной технической системы / В. А. Немтинов, В. Г. Мокрозуб, Е. С. Егоров // Радиотехника. – 2010. – № 12. – С. 21–23.
6. Немтинов, В. А. Оперативное управление транспортными потоками с использованием системы моделирования динамических процессов / В. А. Немтинов, Ю. В. Немтинова, Ю. А. Донских // Вестник компьютерных и информационных технологий . – 2009. – № 3. – С. 21–23.
7. Егоров, С. Я. Автоматизированная информационная система поддержки проектных решений по компоновке промышленных объектов часть 2. Структура и функционирование системы (ч. 1 см. в ИТПП 2009 № 4) / Е. С. Егоров, В. Г. Мокрозуб, В. А. Немтинов, М. С. Громов // Информационные технологии в проектировании и производстве. – 2010. – № 1. – С. 33–39.
8. Немтинов, В. А. Оперативное управление выпуском продукции с использованием системы моделирования динамических процессов / В. А. Немтинов, Ю. В. Немтинова, Д. С. Русских // Вестник Тамбовского государственного технического университета. – 2007. – № 2. – С. 372–378.
9. Малыгин, Е. Н. Автоматизированное проектирование на основе системного подхода / Е. Н. Малыгин, В. А. Немтинов // Экология и промышленность России. – 2001. – № 5. – С. 35–39.

И. Ю. Фролов

Пензенский государственный университет, Пенза, Россия

ОСОБЕННОСТИ ТРАССИРОВКИ ПЕЧАТНЫХ ПЛАТ И ПРИМЕНЕНИЕ БЛОКИРОВОЧНЫХ КОНДЕНСАТОРОВ

Рассматриваются некоторые особенности трассировки печатных плат, которые позволяют обеспечивать устойчивую работу создаваемой схемы.

Трассировка является важным этапом создания печатной платы. Здесь происходит расстановка компонентов схемы и соединение их выходов вручную или с использованием специализированной САПР для проектирования печатных плат. В первом случае человек расставляет элементы схемы самостоятельно, а так же собственноручно проводит трассы между выходами элементов. В случае с применением специализированной САПР автотрассировка выполняется в два этапа, повторяющихся до получения удовлетворительных результатов: размещение компонентов схемы и автоматическая трассировка дорожек.

Основные положения при создании печатной платы: проводники общего провода, а так же питания, следует разводить «звездой» – от одной точки вблизи конденсаторов фильтра питания; создавать проводники минимально возможной длины; сечение проводников питания и общего провода должно быть пропорционально протекающему в них току; в целях упрощения производства компоненты платы часто располагают на одной стороне.

Один из основных методов подавления шума – разделение «земли» на цифровую и аналоговую части. Разделение не подразумевает непосредственно электрической изоляции – стоит создать один узел соединения. Хорошим вариантом будет создание трех отдельных полигонов «земли», объединенных в источнике питания. Стоит учитывать, что автотрассировщики не разделяют «земли».

Нижний слой (англ. "bottom") стоит отводить под наиболее часто подключаемую «землю», а верхний (англ. "top") – под остальные сигналы. Это решение уменьшает наводки и шум, увеличивает механическую прочность печатной платы.

Для снижения уровня помех также необходимо учитывать расположение быстродействующих цифровых элементов.

Углы проводников сглаживаются, чтобы не возникало изменения ширины прохождения сигнала и его отражения. Чем больше сглажен поворот, тем меньше отражения сигнала. Данное действие рекомендуется оставить на заключительный этап разработки платы.

Выбор материала является важным этапом разработки печатной платы, определяющим ее механические свойства. Существует несколько разновидностей материалов, отличающихся диэлектрической проницаемостью, удельным сопротивлением, воспламеняемостью, температурной стабильностью и некоторыми другими свойствами. Наиболее распространенная разновидность – слоистые материалы FR (англ. flame resist, устойчивые к воспламенению). Наиболее распространенными в производстве являются материалы FR-2 и FR-4.

При разработке печатной платы крайне рекомендуется использовать макетирование. Оно применяется при разработке аналоговых схем, в которых, в отличие от цифровых, намного сложнее исправить ошибки на изготовленной плате. Макетирование позволяет существенно сократить время поиска ошибок, а так же предотвратить критические ошибки [3].

Блокировочный конденсатор.

Каждый проводник (например, провод или трасса на печатной плате) обладает индуктивностью, препятствующей изменению тока. Это следует из данного определения индуктивности:

$$U = L \cdot dI / dt.$$

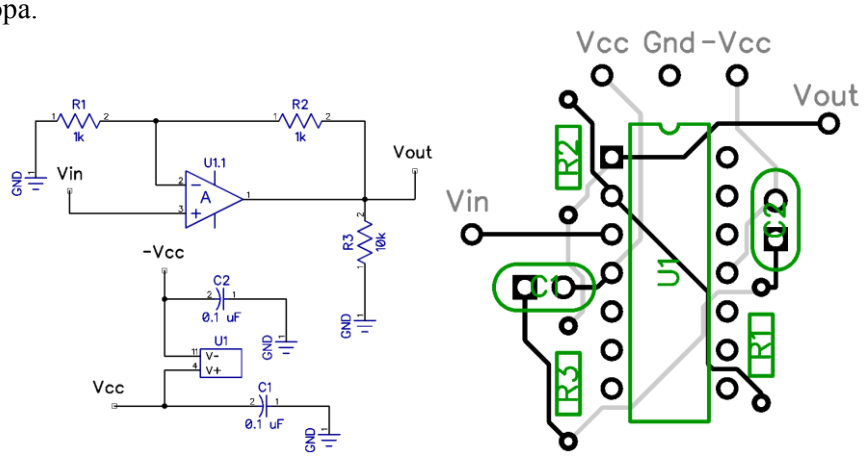
Уравнение показывает, что индуктивность обратно пропорциональна изменению тока. Другими словами, при возрастании индуктивности, сложнее изменяется ток за данный проме-

жуток времени. Чем длиннее проводник, тем больше его индуктивность, и тем медленнее изменяется ток за определенное время. Более того, нарастает падение напряжения [4].

Паразитная индуктивность проводников не позволяет источнику питания обеспечить достаточную мощность микросхемы. Для сведения к минимуму паразитной индуктивности используют блокировочный конденсатор (англ. bypass capacitor). Это конденсатор, шунтирующий питание микросхемы и являющийся местным источником питания [1]. Он устанавливается ближе к выводам питания микросхемы, снижая индуктивность проводников от конденсатора до микросхемы. Предпочтительнее ставить несколько конденсаторов с маленькими емкостями, чем один с большой, так как в этом случае паразитная индуктивность не растет, но емкости складываются [2]. Если применяется несколько конденсаторов разной емкости, то конденсатор с меньшей емкостью должен располагаться ближе к выводу питания из-за своей меньшей индуктивности и лучшего подавления высокочастотных помех.

Если используются микросхемы поверхностного монтажа в крупных корпусах, блокировочные конденсаторы можно разместить на обратной стороне печатной платы. При использовании относительно небольших корпусов емкости не размещаются на противоположной от корпуса стороне, чтобы не увеличивать индуктивность переходных отверстий.

На рис. 1,*а* представлен пример неинвертирующего усилителя с коэффициентом усиления по напряжению 2 [5]. На рис. 1,б предложен вариант печатной платы с установленными блокировочными конденсаторами. Конденсатор C2 установлен вблизи входа питания операционного усилителя. Кроме того, отрицательное напряжение с источника проходит сначала на конденсатор, и только оттуда на вывод питания. Схожим образом подключен и конденсатор C1. Данный прием повышает эффективность блокировочного конденсатора.



а) *б*)

Рис. 1. Применение блокировочных конденсаторов

Заключение

Для создания работоспособной и надежной печатной платы необходимо соблюдать определенные условия при трассировке, такие как корректное размещение проводников и расположение компонентов, применение блокировочных конденсаторов.

Библиографический список

1. Блокировочный конденсатор: описание и использование. – URL: http://chipinfo.pro/article/basics_of_circuitry/blocking_capacitor.shtml (дата обращения: 16.09.17).

2. Фрейман, И. Г. Курс радиотехники / И. Г. Фрейман. – М. : Directmedia, 2014. – 495 с.

3. Аспекты разводки печатных плат. – URL: http://pcbget.ru/Articles/Aspekty_razvodki_PP.html (дата обращения: 16.09.17).

4. Низкочастотные и высокочастотные излучения технических средств: описание паразитных связей. – URL: http://www.intuit.ru/studies/courses/2291/591/lecture/12702?page=2 (дата обращения: 16.09.17).

5. Медведев, А. Сборка и монтаж электронных устройств / А. Медведев. – М. : Litres, 2017. – 105 с.

К. Ю. Шабалова

Пензенский государственный университет, Пенза, Россия

ШАБЛОН АРХИТЕКТУРЫ СИСТЕМЫ «МОДЕЛЬ – ПОВЕДЕНИЕ» ДЛЯ UNITY 3D

Рассматривается принцип разработки приложений на основе архитектурного шаблона «Модель – поведение», его сущность, а также основные компоненты.

Одним из наиболее распространенных и эффективных приемов проектирования программ является использование шаблона программирования MVC (Model-View-Controller) – Модель-Представление-Контроллер. MVC позволяет разделить части программы, отвечающие за хранение и доступ к данным, отображение данных и за взаимодействие с пользователем на отдельные слабо связанные модули. Подобное разделение ответственности упрощает структуру программы и позволяет вносить изменения в одну из этих частей не затрагивая остальные [5].

Шаблон «Модель-поведение» является переработанным шаблоном [3] «Модель-Представление-Контроллер» под специфику игрового движка Unity 3D [2]. Несмотря на то что и сам MVC может быть использован в качестве архитектуры приложения, некоторые его элементы могут носить весьма спорный характер.

В реалиях Unity MVC модель может выглядеть следующим образом (рис. 1).



Рис. 1. Шаблон «Модель-Представление-Контроллер»

В которой компонент Представление отвечает за ввод и вывод информации (в том числе графической) по средствам наследования от класса MonoBehaviour, который предоставляет необходимое API для работы с объектами игровой сцены. Входные данные представлены событиями, выходные – функциями. И, как и полагается этому компоненту, он не выполняет никакой логики.

Компонент модель же, напротив, не является потомком класса MonoBehaviour. Модель отвечает за представление данных об объекте, вроде его координат и здоровья. Доступ к данным обеспечивается через свойства, а при изменении данных происходит вызов событий. Модель выполняет большую часть логики, по принципу «толстая модель, тонкий контроллер».

Контроллер так же не наследуется от MonoBehaviour и является чем-то вроде прослойки между моделью и представлением. Он ожидает входные данные от представления и обновляет модель по мере необходимости. Контроллер так же «прослушивает» изменения данных в модели и выводит их через представление.

Схематично данный шаблон можно изобразить следующим образом (рис. 2).

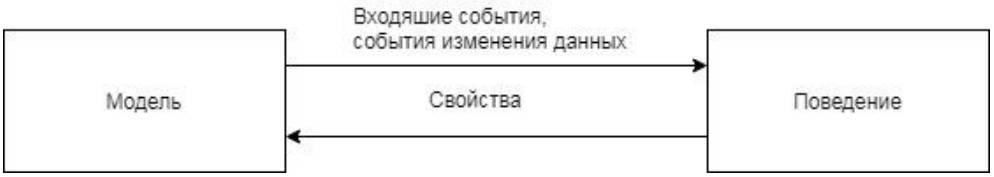


Рис. 2. Шаблон «Модель-Поведение»

Модель и представление объединены в один компонент, названный «модель». Этот компонент – потомок MonoBehaviour, который управляет как вводом-выводом, так и представлением объектов, но в отличие от MVC он не выполняет никакой логики.

Теперь компонент поведение ожидает ввода от модели и соответственно обновляет сами данные в модели. Это может показаться странным, но теперь поведение выполняет всю логику, которая должна выполняться в соответствии со вводом. Так что теперь он не просто передает входные параметры в модель, а сам выполняет логику связанную со вводом. Так же прослушивает изменения данных в модели, выполняет логику, соответствующую этим изменениям и выводит по средствам компонента модели.

Реализация паттерна.

Для примера реализации будет разработан игровой персонаж на языке C# [1]. В первую очередь реализуем интерфейс, через который поведение получает доступ к модели, а так же некоторые классы связанных событий (рис. 3).

```
using System;

using UnityEngine;

// Интерфейс модели
public interface IUnitModel
{
        // Реагирует на изменение здоровья
        event EventHandler<UnitHealthChangedEventArgs> OnHealthChanged;

        // Реагирует на смену позиции
        event EventHandler<UnitPositionChangedEventArgs> OnPositionChanged;

        // Реагирует на отклик по юниту
        event EventHandler<UnitClickedEventArgs> OnClicked;

        // Позиция юнита
        Vector3 Position { get; set; }

        // Здоровье юнита
        float Health { get; set; }

        // Уничтожить объект
        void Destroy();
}

public class UnitClickedEventArgs : EventArgs
{
}
public class UnitHealthChangedEventArgs : EventArgs
{
        public float Health { get; private set; }

        public UnitHealthChangedEventArgs(float health)
        {
                Health = health;
        }
}

public class UnitPositionChangedEventArgs : EventArgs
{
        public Vector3 Position { get; protected set; }

        public UnitPositionChangedEventArgs(Vector3 position)
        {
                Position = position;
        }
}
```

Рис. 3. Реализация интерфейса модели

Следует помнить, что модель включает в себя три части: ввод, вывод и представление данных. События [4] вроде OnClicked для ввода. Мутаторы (setter-ы) вроде того что имеет Potion – для вывода, а свойства вроде Health для представления данных.

Теперь реализуем сам компонент модель через MonoBehaviour (рис. 4 и 5):

```
using System;
using UnityEngine;
// Сочетание модели и представления, но не выполняет никакой логики.
public class UnitModel : MonoBehaviour, IUnitModel
{
        // События данных
        public event EventHandler<UnitHealthChangedEventArgs> OnHealthChanged = (s, e)
=> {};
       public event EventHandler<UnitPositionChangedEventArgs> OnPositionChanged = (s,
e) => {};
        // События входных параметров
        public event EventHandler<UnitClickedEventArgs> OnClicked = (s, e) => {};
        // Данных сохраняются моделью
        // Выводит данные, сопоставляя их с цветом материала
        private float health;
        public float Health
        {
                get { return health; }
                set
                {
                        if (value != health)
                        {
                                health = value;
                                GetComponent<Renderer>().material.color =
Color.Lerp(Color.red, Color.green, value);
                                OnHealthChanged(this, new
UnitHealthChangedEventArgs(health));
                        }
                }
        }
        // Обертка для данных которые уже хранятся в Unity
        // Вывод данных напрямую, без логики
        public Vector3 Position
        {
                get { return transform.position; }
                set
                {
                        if (value != transform.position)
                        {
                                transform.position = value;
                                OnPositionChanged(this, new
UnitPositionChangedEventArgs(value));
                        }
                }
        }
```

Рис. 4. Реализация компонента модель (ч. 1)

Модель заботится о таких деталях, как «как узнать когда игрок кликнул по персонажу» и «как отрендерить здоровье». Так же есть вариант перемещения части кода, ответственной за отрисовку здоровья в цвет и добавить свойство setColor в модели для вызова поведением. Некоторые могут предпочти перенос всей логики в поведение, а другие оставить код рендеринга в модели. Оба варианта приемлемы, оба допустимы, поэтому в данном примере представлен вариант с рендерингом в модели.

Так же следует отметить, что модель не имеет явной связи с поведением. Она не знает какой класс отвечает за логику. модель просто знает что произошел ввод а значит нужно вызвать события, функции вывода и установить вызванные свойства. Тут так же хранятся данные, которые уже хранит Unity, вроде transform.position. Более важно то, что тут не выполняется никакой логики, кроме захвата ввода и рендеринга.

Далее реализуем поведение (рис. 6).

```
// Установка начальных значений
        void Awake()
        {
                Health = 1;
        }
        // Обработка ввода путем отправки события, не выполняющего логику
        void Update()
        {
                if (Input.GetMouseButtonDown(0))
                {
                        var ray = Camera.main.ScreenPointToRay(Input.mousePosition);
                        RaycastHit hit;
                        if (Physics.Raycast(ray, out hit) && hit.transform ==
transform)
                        {
                                OnClicked(this, new UnitClickedEventArgs());
                        }
                }
        }
        // Уничтожение юнита
        public void Destroy()
        {
                Destroy(gameObject);
        }
}
```

Рис. 5. Реализация компонента модель (ч. 2)

```
public class UnitBehaviour
{
        private readonly IUnitModel model;
        private float clickDamage;
        public UnitController(IUnitModel model, float clickDamage)
        {
                this.model = model;
                this.clickDamage = clickDamage;

                // Ожидает входных данных от модели
                model.OnClicked += HandleClicked;

                // Ожидает изменения в модели
                model.OnHealthChanged += HandleHealthChanged;
        }
        private void HandleClicked(object sender, UnitClickedEventArgs e)
        {
                // Выполнение логики как результат ввода
                // Простые подсчеты нанесения урона
                model.Health -= clickDamage;
        }
        private void HandleHealthChanged(object sender, UnitHealthChangedEventArgs
e)
        {
                // Выполнение логики как результат ввода
                // Если у цели нет здоровья - объект уничтожается
                // И выводит данные в модель
                if (e.Health <= 0)
                {
                        model.Destroy();
                }
        }
}
```

Рис. 6. Реализация поведения

Стоит отметить, что поведение не является потомком MonoBehaviour и не имеет никакой связи с ним. Он использует модель через IUnitModel.

Далее следует отметить, что поведение не хранит никаких данных о персонаже вроде здоровья и координат, для этого он использует модель. Он так же использует события модели для ожидания ввода и изменений в представлении данных. Поведение выполняет всю логику: вычисления урона, хоть и очень простого, а так же сообщает игре, что когда у

персонажа не остается жизней – он умирает. Поведение использует модель для вывода результатов подсчета урона и перехода персонажа в состояние смерти.

Заключение

Пример данного шаблона полностью работоспособен, и на его основе может происходить дальнейшая разработка приложения, при этом будет сохраняться отличная читаемость кода, ввиду разграничения бизнес-логики и данных.

Библиографический список

1. Фримен, А. ASP.NET MVC 5 с примерами на C# 5.0 для профессионалов / А. Фримен. – М., 2015. – 684 с.

2. Паласиос, Х. Unity 5.x. Программирование искусственного интеллекта в играх / Х. Паласиос. – М. : ДМК Пресс, 2016. – 272 с.

3. Шпаргалка по шаблонам проектирования: описания популярных паттернов. – URL: https://habrahabr.ru/post/210288/ (дата обращения: 16.09.17).

4. Events: события в Unity 3D. – URL: https://unity3d.com/ru/learn/tutorials/topics/scripting/events (дата обращения: 16.09.17).

5. Model-View в QML. Часть нулевая, вводная. – URL: https://habrahabr.ru/post/181712/ (дата обращения: 16.09.17).

УПРАВЛЕНИЕ В СОЦИАЛЬНЫХ, ЭКОНОМИЧЕСКИХ И ТЕХНИЧЕСКИХ СИСТЕМАХ

В. С. Мкртчян[1], Л. А. Гамидуллаева[2]

[1]HHH University, Сидней, Австралия
[2]Пензенский государственный университет, Пенза, Россия

МЕХАНИЗМЫ ЭФФЕКТИВНОГО ЦИФРОВОГО ВЗАИМОДЕЙСТВИЯ УЧАСТНИКОВ В ИННОВАЦИОННОЙ СИСТЕМЕ РЕГИОНА

Исследования и разработки стремительным образом приобретают глобальный характер и осуществляются, преимущественно, на коллаборационной основе, что обусловливает необходимость поиска партнеров, ресурсов и идей за пределами компании. Благодаря активному развитию информационных технологий в международной практике стали появляться новые, не нашедшие еще применения в России, виды инновационных посредников, которые могут быть адаптированы к работе в российской инновационной системе. Статья посвящена анализу современных передовых видов регионального инновационного посредничества, основанных на цифровых технологиях, и поиску возможностей их адаптации в России.

Введение. На сегодняшний день необходимость цифровизации российской экономики уже не вызывает сомнений и рассматривается как краеугольный камень всей экономической политики государства, обеспечивающий достижение Россией глобальной конкурентоспособности и национальной безопасности [14]. На наш взгляд, цифровизация экономики РФ позволит также устранить главный барьер, препятствующий экономической активности субъектов, - высокие трансакционные издержки, снижающие уровень взаимодействия участников и не позволяющие в полной мере реализовать потенциал социально-экономической системы. При этом, несмотря на то что Правительством РФ утверждена Программа развития цифровой экономики (от 28.07.2017 № 1632-р.), следует признать отсутствие конкретных механизмов реализации социально-экономических процессов на основе цифровых технологий.

Экономический анализ цифровой экономики уже около двух десятилетий является активно развивающейся областью научных исследований, начиная с введения самого понятия «цифровая экономика» Н. Негропонте [12]. Согласно международному исследованию, проведенному авторами [7], экономическое развитие, информационно-коммуникационная инфраструктура и государственное управление являются ключевыми составляющими цифрового развития страны. Российских научных трудов по данной проблематике гораздо меньше, посвящены они исследованию передового опыта в области цифровой экономики, анализу инструментов, моделей и стандартов функционирования умных городов, системы безопасности, транспорта, здравоохранения на основе принципов и механизмов кибер-управления [1–5]. Проблеме взаимодействия участников инновационного процесса, на наш взгляд, не уделяется достаточного внимания в современных исследованиях. В РФ сегодня функционирует огромное число посредников инноваций (бизнес-инкубаторы, технопарки, инновационные центры и т.д.). Однако их эффективность по-прежнему признается крайне низкой.

Подход Living Labs («Живых лабораторий») для инновационного развития региона. Стремительное развитие информационных технологий привело к появлению новых форм инновационного посредничества, в частности «Живых лабораторий», которые облегчают диффузию технологий, интеллектуальной собственности и лицензирования в иннова-

ционной системе [6]. Авторы [8] выделяет около десяти основных функций посредников в сфере инноваций, которые остаются малоизученными в литературе. Существует необходимость проанализировать возможности и ограничения реализации данных функций в контексте региональных инновационных систем в России.

На сегодняшний день по всему миру насчитывается более сотни Живых лабораторий. Данные виды посредничества создавались, в основном, как государственно-частные партнерства для реализации потенциальных преимуществ региона, где движимые потребителями инновации интегрированы в совместный процесс создания новых сервисов, продуктов и инфраструктуры. Это форма инновационного посредничества, ориентированная на создание функциональной сферы, где все заинтересованные стороны образуют государственно-частное партнерство между университетами, государственными ведомствами, институтами и обществом с целью создания, прототипирования, проверки новых технологических продуктов и режиме реального времени. В результате появляется общая платформа для ускорения инновационного процесса, обеспечивается оказание средне- и долгосрочных услуг для развития новых технологий, поддержки инновационного процесса различных организаций.

Особенности «Живых лабораторий» как вида инновационных посредников состоят в следующем:

1. *Способствуют сотрудничеству в сфере исследований*. «Живые лаборатории» функционируют как коннекторы, ищущие технологические комплементарности и материализующие связи на этой основе. При этом они способствуют сотрудничеству различных организаций путем проведения средне- и долгосрочных исследований возможных видов технологий с различными акторами – заинтересованными сторонами, а также путем вовлечения потребителей на этапе проведения исследований и разработок технологий.

2. *Оказание различных услуг для заинтересованных сторон*. Как правило, центры исследований и разработок постоянно расширяют спектр предоставляемых услуг по пути включения дополнительных видов деятельности, например, обоснование проектов, проверка, маркетинговый анализ и др. Отличие Живых лабораторий состоит в том, что они предлагают комплементарные сервисы, не только создание технологий и разработку, но также предлагают экспериментальную платформу с большим количеством пользователей, которые вовлечены в совместный изобретательский процесс через использование прототипов инновационной продукции.

3. *Связь науки и государства.* Данные посредники способствуют повышению результативности деятельности отдельных регионов, поэтому инициаторами их создания в основном являются университеты или органы государственной власти. Частных таких лабораторий пока еще очень мало. Подобные инициативы направлены на ускорение развития новых технологий в регионе и представляют собой государственные агентства, которые активно продвигают направление исследований и создают синергетические эффекты между региональными акторами.

Взаимодействие компаний в Живых лабораториях происходит посредством использования специальных сервисов поиска знаний, позволяющих искать новые решения в сети внешних акторов, заинтересованных в сотрудничестве.

Процесс присвоения знаний состоит из шести основных этапов: идентификация потребности, представление (триангуляция) потребности, спецификация потребности, поиск и сбор информации, оценка, выбор решения [9].

Заключение. На наш взгляд, инновационные посредники должны рассматриваться как ключевой фактор в рамках региональных инновационный систем. «Живые лаборатории» как современный тип инновационных посредников действуют между основными участниками инновационной системы, создают необходимые связи, формируют возможности для развития взаимоотношений и кооперативного поведения участников. Они выполняют функции трансфера знаний, организации кооперации между наукой и реальной экономикой, координации спроса, производства знаний и коллаборации, коммерциализации знаний, прогнозирований будущих тенденций, стандартизации и регулирования, тестирования и оценки и др.

Таким образом, «Живые лаборатории» представляют собой передовой тип инновационного посредничества, который необходимо внедрить и апробировать в рамках инновационных систем субъектов РФ.

Библиографический список

1. Умные города: модели, инструменты, рэнкинги и стандарты / В. И. Дрожжинов, В. П. Куприяновский, Д. Е. Намиот, С. А. Синягов, А. А. Харитонов // International Journal of Open Information Technologies. – 2017. – №3.
2. Куприяновский, В. П. Принятие решений в цифровой экономике. Опыт Великобритании / В. П. Куприяновский, С. Н. Евтушенко, О. Н. Дунаев, В. И. Дрожжинов, Д. Е. Намиот // International Journal of Open Information Technologies. – 2017. – № 4.
3. Куприяновский, В. П. Умная полиция в умном городе / В. П. Куприяновский, С. А. Буланча, Д. Е. Намиот, С. А. Синягов // International Journal of Open Information Technologies. – 2016. – № 3.
4. Курганов, В. М. Транспортные системы и комплексы в ретроспективе развития / В. М. Курганов // Вестник СибАДИ. – 2016. – № 1 (47).
5. Чеклецов В. В. Россия – Германия: повышение конкурентоспособности российской и германской экономик. Сотрудничество в рамках концепции «Индустрия 4. 0» (обзор «Круглого стола» на Петербургском международном экономическом форуме – 2016) / В. В. Чеклецов // Философские проблемы информационных технологий и киберпространства. – 2016. – № 1 (11).
6. Benassi, M. Playing in between: patent brokers in markets for technology / M. Benassi, Di A. Minin // R&D Management. – 2009. – № 39 (1). – P. 68–86.
7. Chinn, M. D. ICT use in the developing world: an analysis of differences in computer and Internet penetration / M. D. Chinn, R. W. Fairlie // IZA Discussion Paper. – 2006. – № 2206.
8. Howells, J. Intermediation and the role of intermediaries in innovation / J. Howells // Research Policy. – 2006. – № 35 (5). – P. 715–728.
9. Lopez-Vega, H. Open innovation: Organizational practices and policy implications / H. Lopez-Vega. – Hasselt University, 2013.
10. Handbook of Research on Estimation and Control Techniques in E-Learning systems / V. Mkrttchian, A. Bershadsky, A. Bozhday, M. Kataev, S. Kataev ; eds. Hershey. – PA, USA: IGI Global, 2016. – 679 p.
11. Mkrttchian, V. Use Online Multi-Cloud Platform Lab with Intellectual Agents: Avatars for Study of Knowledge Visualization & Probability Theory in Bioinformatics / V. Mkrttchian // International Journal of Knowledge Discovery in Bioinformatics. – 2015. – № 5 (1). – P. 11–23.
12. Negroponte, N. Being Digital / N. Negroponte. – NY : Knopf, 1995. – 256 p.
13. Vasin, S. M. State support for innovative entrepreneurship as a factor of ensuring national security during an economic crisis / S. M. Vasin, L. A. Gamidullaeva // Trends and Management, I. – 2016. – № 2 (2). – P. 137–144.
14. Vasin, S. M. About Russian Regional Users' Innovation Based on Digital Information / S. M. Vasin, L. A. Gamidullaeva // User Innovation and the Entrepreneurship Phenomenon in the Digital Economy / In Pedro Isaias and Luisa Carvalho (Eds). – Hershey, PA, USA: IGI Global, 2017. – P. 301–330.

М. В. Арестова, Л. Р. Фионова

Пензенский государственный университет, Пенза, Россия

АНАЛИЗ ПРОЦЕССОВ ИСПОЛЬЗОВАНИЯ ЭЛЕКТРОННОЙ ПОДПИСИ НА ПРОМЫШЛЕННОМ ПРЕДПРИЯТИИ

Проанализировано использование электронной подписи на промышленном предприятии как элемента электронного документооборота. Выявлены проблемы использования электронной подписи, связанные с отсутствием регламентирующих документов. Предложены пути решения.

В связи с широким распространением информационных технологий (ИТ) во всех сферах деятельности современного общества управление различными процессами на предприятиях не является исключением. Важную роль в данной области играют системы элек-

тронного документооборота (СЭД), которые предназначены для организации и автоматизации процессов взаимодействия между сотрудниками предприятия. Использование данных систем обеспечивает эффективное управление документами и увеличивает производительность работы сотрудников предприятия [1].

В АО «Сердобский механический завод» («СМЗ») достаточно большой объем документов. На предприятии существует и традиционный (бумажный – 841 шт./год) и электронный документооборот (967 шт./г.).

Для автоматизации деятельности АО «СМЗ» используется в первую очередь программное обеспечение: «1С: Предприятие».

Оно состоит из двух частей:

1. Платформа.
2. Прикладное решение.

Прикладными решениями являются следующие программы:

1. Документооборот.
2. Управление производственным предприятием.
3. Бухгалтерия.
4. Отчетность.
5. Зарплата и управление персоналом.

Рассмотрим более подробно прикладное решение «1С: Документооборот».

При запуске «1С: Документооборот» открывается стартовая страница, на которой сразу можно посмотреть входящие документы и задачи. Далее можно рассмотреть входящие, а так же исходящие и внутренние документы более подробно.

Виды документов, которые создаются в «1С: Документооборот»:

- приказы по основной деятельности (174 шт./г.);
- приказы по личному составу (258 шт./г.);
- приказы-постановления (117 шт./г.);
- договоры (102 шт./г.);
- протоколы (75 шт./г.);
- распоряжения (77шт./г.);
- распоряжения по личному составу (51 шт./г.);
- указания (12 шт./г.);
- служебные записки (101шт./г.).

Каждый документ из списка можно посмотреть. Так же можно посмотреть некоторую информацию о документе: его реквизиты; резолюции; связи; визы; рабочую группу. При переходе к исходящим документам, формируется соответствующий список документов. При просмотре исходящего документа можно внести некоторые изменения и посмотреть информацию о данном документе. Меню программы позволяет просмотреть все созданные пользователем документы. Но в связи с тем, что на предприятии нет нормативных актов, регламентирующих работу с электронными документами в СЭД, все операции освоены лишь некоторыми сотрудниками. Обязанности, связанные с использованием СЭД не прописаны ни в одной должностной инструкции. Каждый пользователь приобретает и накапливает свой опыт.

Важным элементом электронного документооборота является электронная подпись (ЭП). Ее применение необходимо для организации полноценного электронного документооборота на предприятии.

Можно выделить главную задачу ЭП – установить авторство документа и гарантировать, что документ не был изменен после его подписания [2].

В результате проведенного анализа было выявлено, что ЭП на предприятии применяется в следующих областях:

1. Электронный документооборот.
2. Электронная отчетность для контролирующих органов.
3. Электронные торги.
4. Обмен сообщениями с помощью средств электронной почты.
5. Оформление трудовых отношений.

Рассмотрим более подробно применение ЭП в вышеуказанных ситуациях.

Самые распространенные документы внутреннего документооборота АО «СМЗ» – это служебные записки и приказы по основной деятельности.

При бумажном документообороте автор служебной записки вынужден самостоятельно пройти всех заинтересованных лиц и собрать нужные подписи, возможно, вплоть до конкретного исполнителя. При использовании СЭД, инициатор может создать документ, подписать его ЭП и отправить руководителю. В свою очередь, руководитель может отправить документ дальше на исполнение, либо выпустить и подписать отдельный документ, связанный со служебной запиской. Таким образом, инициатор в любой момент времени может отследить текущее состояние по своему документу.

При составления любого вида документа внутри организации используется простая ЭП, так как здесь важно только авторство документа.

Отчетность осуществляется через электронные каналы связи в виде регламентированных форм подписанных ЭП, зашифрованных (при помощи дополнительно устанавливаемой на компьютере пользователя программы-криптопровайдера) и, впоследствии, отправленных в контролирующие государственные органы (ФНС, ФСС, ПФР). В этом случае используется программа «1С-Отчетность».

Подготовкой и отправкой электронных документов для ФНС занимается отдел бухгалтерского учета и контроля. Документы отправляются из «1С: Отчетность» и подписываются усиленной квалифицированной подписью. Отправка документов так же может происходить через личный кабинет юридического лица на официальном сайте ФНС России. Для того чтобы использовать личный кабинет обязательно наличие усиленной квалифицированной ЭП.

Отдел бухгалтерского учета и контроля так же занимается подготовкой и отправкой электронных документов в ФСС и ПФР. Это происходит, как уже отмечалось, через сервис «1С: Отчетность». Этот сервис встроен в программу «1С: Бухгалтерия», что является очень удобным для сотрудников отдела. Многие из них самостоятельно, как смогли, освоили работу с данной программой. Инструкций и регламентирующих документов нет.

Отправка документов в ПФР осуществляется через личный кабинет плательщика на официальном сайте ПФР. Для входа в личный кабинет используется регистрационный номер АО «СМЗ» в ПФР. Отправка документов связана с использованием усиленной квалифицированной ЭП.

Все закупки, выбор поставщиков, подрядных организаций осуществляются посредством электронных торгов. Проведение электронных торгов сопровождается рядом документов, которые подтверждаются ЭП. Электронные торги проходят на Электронной торговой площадке. При этом используется усиленная квалифицированная ЭП, она обязательна для входа на сайт и дальнейшей работы с ним.

Одним из часто создаваемых документов, подписываемых ЭП, является договор. Рассмотрим процесс создания договора. Сотрудник отдела закупок АО «СМЗ» готовит проект договора и далее отправляет его на согласование. Руководитель вносит изменения в договор, подписывает его и отправляет сотруднику – исполнителю. В связи с тем, что изменений бывает много, использование ЭП помогает сэкономить время и средства.

Далее этот договор отправляется компании-контрагенту. Если его устраивают условия договора, то он подписывает его и направляет в АО «СМЗ». В данном случае может использоваться как усиленная квалифицированная ЭП, так и простая ЭП. Опять же нет регламентирующего документа, определяющего, когда и какую ЭП, использовать.

При обмене документами с удаленными предприятиями в АО «СМЗ» обычно используется корпоративная электронная почта. В данном случае применяется ЭП для того, чтобы в первую очередь гарантировать авторство документа. Процедура использования ЭП такая же, как и при внутреннем электронном документообороте. Соответственно используется простая ЭП.

На предприятии еще одним случаем применения ЭП является оформление трудовых отношений с работниками, которые работают удаленно. Изменения в Трудовом кодексе РФ регулируют труд дистанционных работников и предусматривают заключение трудового договора о дистанционной работе путем обмена электронными документами. В таком случае используются усиленные квалифицированные ЭП дистанционного работника или

лица, поступающего на дистанционную работу, и работодателя [3]. Подготовкой таких документов занимаются сотрудники бюро по работе с персоналом в программе «1С: Зарплата и управление персоналом». В других случаях работы с внутренними электронными документами сотрудники бюро используют простую ЭП.

Таким образом, переход от бумажного документооборота к электронному связан с использованием ЭП. Но для получения конкретного положительного результата, - необходимо принятие и освоение этого новшества всеми сотрудниками предприятия. Стоит отметить, в первую очередь, сложность и запутанность использования различных видов ЭП. При этом проблемным моментом является отсутствие достаточной правовой регламентации ответственности всех участников электронного взаимодействия. Важным шагом для решения этой проблемы будет установление такой ответственности работников предприятия локальными актами.

Внедрению ЭП должно в обязательном порядке предшествовать обследование предприятия для выяснения конкретных задач, которые нужно сопровождать электронными документами, заверенными (скрепленными) ЭП соответствующего вида и сразу же разработка полного комплекта нормативно-методического обеспечения внедрения ЭП, внесение изменений в должностные инструкции и регламенты, обучение персонала.

Библиографический список

1. Фионова, Л. Р. Опыт внедрения СЭД в деканате вуза / Л. Р. Фионова, О. В. Миронова // Современные технологии документооборота в бизнесе, производстве и управлении : сб. ст. XV Междунар. науч.-практ. конф. – Пенза : Приволжский Дом знаний, 2015. – С. 88–92.

2. Об электронной подписи : федер. закон от 06.04.2011 № 63-ФЗ.

3. Трудовой кодекс Российской Федерации от 30.12.2001 № 197-ФЗ (с изм. и доп., вступ. в силу с 01.01.2017).

Л. Р. Фионова, С. О. Шибаева

Пензенский государственный университет, Пенза, Россия

РАЗРАБОТКА БАЗЫ ДАННЫХ ДЛЯ АВТОМАТИЗИРОВАННОГО СОЗДАНИЯ ДОЛЖНОСТНЫХ ИНСТРУКЦИЙ

Описан процесс разработки базы данных для автоматизированного создания должностных инструкций. Они должны в настоящее время создаваться на основе профессиональных стандартов. Это требование законодательства. Описанный подход позволяет ускорить подготовку должностных инструкций на все должности, входящие в конкретный профессиональный стандарт.

В настоящее время продолжается процесс внедрения профессиональных стандартов в практику работы кадровых служб. В связи с принятием Федерального закона «О независимой оценке квалификации» кадровые службы должны привести должностные инструкции (ДИ) сотрудников в соответствие с профессиональными стандартами (ПС) [1]. Для ускорения этого процесса актуальной становится задача разработка базы данных, которая бы позволила автоматизировать подготовку ДИ.

Для ускорения сбора и последующей обработки данных предлагается использовать программные средства *Microsoft Access*. Этот выбор обоснован тем, что пакет *Microsoft Office* является достаточно популярным. База данных (БД) *Access* удобна и проста в работе. Работая в *Access*, можно открывать и редактировать файлы с помощью кнопок, команд и клавиш, а также использовать буфер обмена *Office* для копирования данных из таких программ, как *Microsoft Word* или *Microsoft Excel*. Это позволяет легко получить требуемые отчеты и диаграммы для анализа, возможно автоматическое формирование определенных

видов документов на основе существующих данных в таблицах, формах и запросах. Это удобно, так как сокращает время и трудозатраты на ручную подготовку документов, уменьшает риск совершения ошибок в заполнении и оформлении документа.

Экранные формы проектируются на основе таблиц, представленных в ПС. Разработать структуру БД – значит определить, какие таблицы входят в БД и из каких атрибутов состоит каждая из них, а также выяснить, каким образом таблицы БД связаны между собой.

На начальном этапе создадим БД только для разработки ДИ делопроизводителя, но предусматривающую дальнейшее расширение для создания ДИ сотрудников отдела документационного обеспечения управления (ДОУ) и кадров, а также возможность разработки на их основе положений о подразделениях.

В качестве правильных объектов берем ключевые объекты, используемые при составлении ДИ, и являющиеся переменными в ней – должность, ДИ (формируемый в дальнейшем документ), разделы документа – отдельно «Задачи и функции», «Должностные обязанности», «Права», «Ответственность», «Взаимодействия и связи по должности». Для заполнения раздела «Общие положения» введем еще два объекта: знания должностного лица и умения. В разрабатываемой БД все объекты представим в виде таблиц.

Каждому объекту назначается ID – первичный ключ, внешние ключи для связи с другими объектами БД, а затем и другие свойства объекта, представляющие собой смысловую составляющую документа. Введем общие ограничения целостности при создании таблиц и формировании структуры БД.

Далее на основе каждой таблицы создаем форму для заполнения конкретных данных конечным пользователем.

В форме «Должность» находятся поля № дол. – ключевое поле таблицы; наименование должности – вводится согласно профессиональному стандарту сотрудника отдела ДОУ [2]; категория – по стандарту; подразделение, должность руководителя подразделения – согласно организационной структуре конкретной организации или штатному расписанию; кому подчиняется – указывается должность соответствующего сотрудника в дательном падеже; ФИО руководителя – указываются инициалы и фамилия руководителя. Все перечисленные поля нужны для корректного заполнения раздела «Общие положения» ДИ.

Кроме того, для заполнения «Общих положений» используются данные из форм «Умения должностного лица» и «Знания должностного лица». В данных формах содержатся поля с уже заполненными умениями и знаниями, которые берутся для конкретного должностного лица из ПС [2, 3] (в соответствии с его трудовыми функциями). По желанию составитель может дополнить данный раздел.

Форма «Функции» предназначена для заполнения раздела должностной инструкции «Задачи и функции». Поля «должность», «трудовые функции» и «обобщенная трудовая функция» показывают связь между обозначенными категориями ПС и заполняются в соответствии с ним.

Должностные обязанности специалиста отдела ДОУ соответствуют трудовым действиям, установленным ПС [2, 3] . Все они указаны в форме «Должностные обязанности». Составителю лишь требуется отметить галочкой те из них, которые соответствуют специфике работы специалиста в конкретной организации и войдут в его ДИ.

Положения четвертого раздела «Права» не регламентированы ПС, поэтому заполняются полностью самостоятельно пользователем для каждой конкретной должности. Поле «Права», следует заполнять как законченное предложение, т.е. с заглавной буквы и с точкой на конце – для корректного формирования документа Должностная инструкция. При формулировании прав следует опираться на выполняемые сотрудником должностные обязанности, которые также отражены в представленной форме. Для помощи пользователю отдельно создается подчиненная форма «Должностные обязанности делопроизводителя», к которой конечный пользователь не имеет доступа.

Аналогично выглядит и функционирует форма «Ответственность». Ответственность должностного лица возникает за неисполнение или ненадлежащее исполнение своих должностных обязанностей. Формулировать положения раздела следует, опираясь на должностные обязанности лица. Кроме того в рабочем процессе существуют такие нару-

шения, за которые любой сотрудник несет ответственность. Например, за нарушение правил внутреннего трудового распорядка, трудовой дисциплины, правил техники безопасности и противопожарной безопасности.

Форма «Взаимодействия» в конечном итоге формирует соответствующий раздел в ДИ. В поле «с кем и как взаимодействует» указываются наименования должностей и подразделений, с которыми у должностного лица, выбранного в поле «должность», в процессе выполнения своих должностных обязанностей образуются деловые связи. Для облегчения работы пользователя стоит сразу внести основные взаимодействия делопроизводителя, но составитель имеет возможность при необходимости добавить свои пункты.

Помимо основных разделов, должностная инструкция содержит «Регистрационный номер», «Дату», «Гриф утверждения», «Подпись», «Ознакомительную визу», которые позволят корректно сформировать реквизиты заголовочной и оформляющей части документа. Для их заполнения создается форма «Должностная инструкция». Во всех создаваемых формах максимально обеспечивается удобство пользования БД путем ввода частных ограничений целостности данных (поля со списком, всплывающие подсказки) и удобства интерфейса (контрастный текст, вспомогательные окна, кнопки управления формой, ссылка на ПС, формирование и печать документа). Такие меры защищают пользователя от совершения ошибок при заполнении БД, сокращают затраты времени на работу с СУБД. Пояснительные тексты на формах, подсказки, разработанное по окончании работы руководство пользователя позволяют пользователю корректно заполнить все данные и сформировать документ, правильный с точек зрения делопроизводства и грамматики.

После заполнения всех форм можно приступать к работе над формированием самого документа.

Необходимо создать запросы, чтобы связать должность, на которую разрабатывается ДИ, с данными, заполняемыми через формы «Права», «Ответственность», «Знания», «Умения» путем суммирования по полю №дол и выборке по должности. Запрос «Должностная инструкция» создается для того, чтобы собрать воедино основные части будущего документа путем сложения, выборки и наложения на поля условий. Так, запросом объединяются данные таблиц «Должность», «Трудовые действия», «Должностная инструкция». При формировании ДИ делопроизводителя на поле наименование должности ставим условие отбора «делопроизводитель». При расширении БД данное условие заменяется на окно ввода информации с текстом «введите должность, на которую формируется должностная инструкция». На поле «включить в документ» накладывается условие «=Истина», чтобы в документе отражались только те трудовые действия, которые пользователь отметил галочкой на форме «Трудовые действия». Вид созданного запроса в режиме Конструктор представлен на рис. 1.

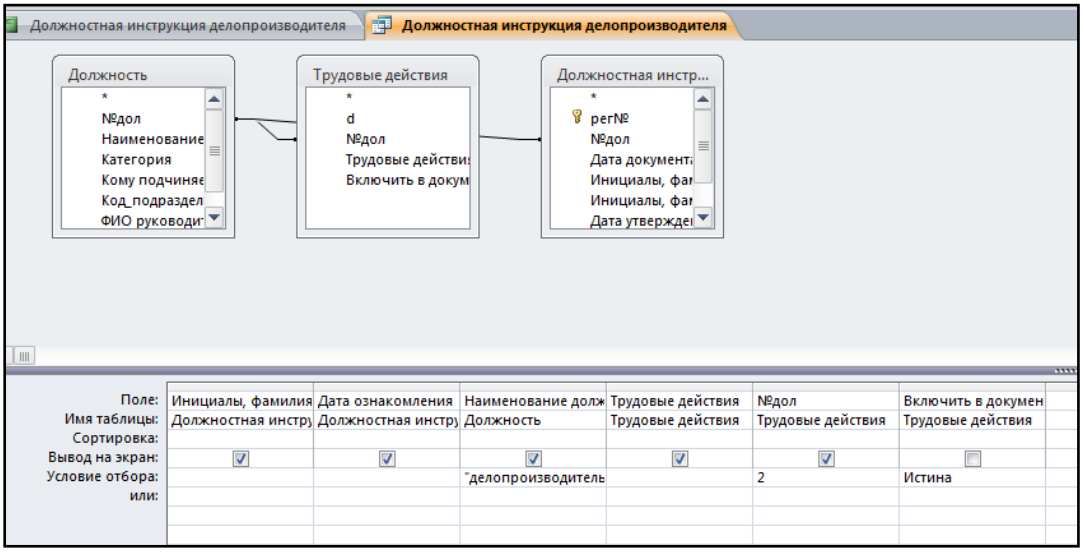


Рис. 1. Запрос «Должностная инструкция делопроизводителя»

Далее формируется отчет на каждый созданный запрос. Создаваемый документ Должностная инструкция будет базироваться на соответствующем отчете по запросу. Остальные отчеты будут вставляться в основной через инструмент «Подчиненная форма/отчет».

При использовании всех сведений из ПС для формирования вышеописанных форм можно создать ДИ на любую должность, описанную в ПС.

ПС, по сути, является фундаментальным документом, позволяющим работодателю находить «точки опоры» при разработке важнейших организационных документов, которыми являются ДИ. Автоматизация этих процессов позволяет повысить эффективность управления персоналом.

Библиографический список

1. О независимой оценке квалификации : федер. закон № 238-ФЗ от 03.07.2016.
2. Специалист по организационному и документационному обеспечению управления организацией : профессиональный стандарт [утв. приказом Минтруда и соцзащиты РФ № 276н от 06.05.2015).
3. Специалист по управлению документацией организации : профессиональный стандарт [утв. приказом Минтруда и соцзащиты РФ № 416н от 10.05.2017].

П. К. Чечель, А. А. Краснов, Д. А. Трокоз

Пензенский государственный университет, Пенза, Россия

РАЗРАБОТКА КОНЦЕПТА ИНТЕРАКТИВНЫХ КАРТ КРИМИНАЛИЗАЦИИ РАЙОНОВ НА ОСНОВЕ ОФИЦИАЛЬНЫХ ДАННЫХ В ОТКРЫТЫХ ИСТОЧНИКАХ

Рассматриваются способы сбора статистической информации, ее обработки и публикации в свободном доступе.

Проблема выбора места для переезда семьи или передислокации бизнеса всегда была сложной в силу множества критериев, которые зачастую не имеют четкой оценки. Одним из первостепенных критериев является безопасность. Ежегодно генеральной прокуратурой Российской Федерации составляются и публикуются статистические, а также аналитические обзоры и рейтинги преступлений. При этом, публикуемые данные слишком обобщенные (наименьшим элементом со своими статистическими показателями является субъект Российской Федерации), и, тем самым, неточно отражают действительное положение дел. В рамках данной проблемы, актуальной задачей является также визуализация данных в виде картографического сервиса.

Генеральная прокуратура Российской Федерации разработала портал правовой статистики [1], где в открытом доступе находятся ключевые показатели преступности в стране и регионах. На данный момент портал находится в режиме опытной эксплуатации. Однако уже сейчас на сайте размещена карта страны с градацией регионов по количеству преступлений. Но такая статистика не учитывает население регионов, а значит закрашенные темными цветами регионы (рис. 1), как правило, обладают высокой численностью населения, а не удручающей криминальной обстановкой. Для анализа безопасности того или иного региона необходимо взять общее число населения региона и разделить на общее количество преступлений, совершенных в этом регионе. Чем большее число получится в итоге – тем безопаснее регион [2].

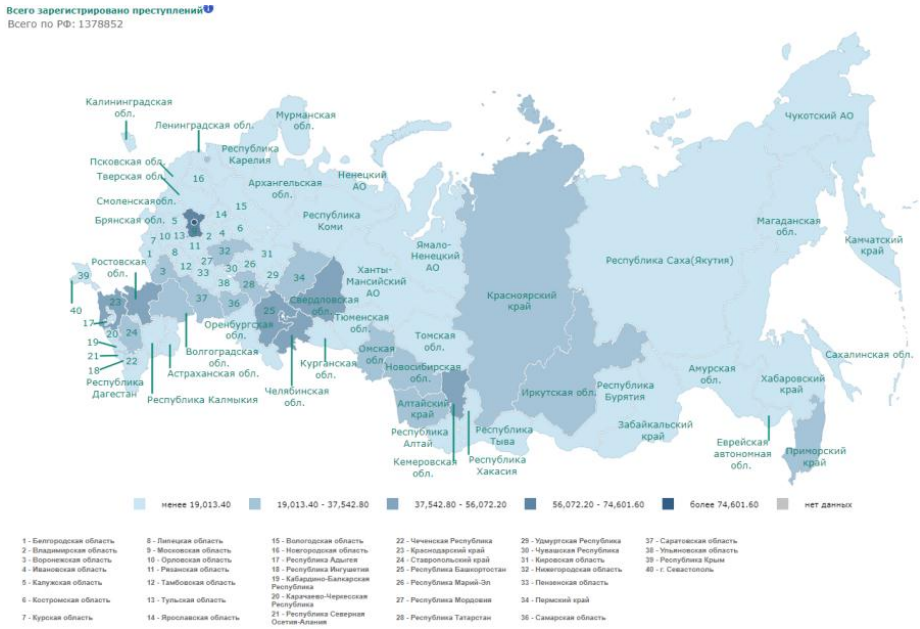


Рис. 1. Страница портала правовой статистики

Предполагается размещение карт городов, с делением по районам, данный уровень масштаба помогает более точно оценить криминальную обстановку. Для этого выделяются контуры (границы) города и районов, а площадь (все пространство внутри контура) района заливается цветом, отображающим уровень преступности. Проще всего задействовать для таких целей интуитивно понятный набор цветов. Если использовать десятибалльную шкалу (очень безопасный район – единица, сверх криминализированный район – десять), то достаточно будет гаммы перехода от зеленого к красному цвету (через желтый). Цветовая информация с указанием шкал может дублироваться в виде легенды рядом с картой города. Данная идея была впервые реализована пользователями социальной сети. Пользователи создали карту-схему Саратова с делением на районы, каждый район был залит определенным цветом, определяющим степень “опасности” данного района. При составлении карты пользователи руководствовались субъективными ощущениями и собственным мнением, не опираясь на официальную статистику. После публикации карты Саратова, пользователи из других регионов стали составлять и публиковать подобные карты, главной целью которых было показать наиболее и наименее криминализированные районы города.

В качестве фильтров данных крайне удобно использовать временные промежутки различной длины. Так, пользователи смогут посмотреть разницу между текущим положением криминализации и статистикой за длительный промежуток времени. Имея предложенные инструменты фильтрации удобно отслеживать тенденции понижения или повышения уровня преступности для выбранного района или города целиком.

Для этого необходимо реализовать следующие фильтры отражающие статистические показатели:

- за месяц;
- за квартал;
- за полугодие;
- за год;
- за 3 года;
- за 5 лет.

Раздел новостей целесообразнее всего посвятить самым «громким» криминальным трагедиям в регионах. Создание раздела с набором советов, инструкций и рекомендаций, как обезопасить себя и своих близких, тоже является актуальным и полезным функционалом в рамках реализации описываемого проекта.

К сожалению, для реализации интерактивных карт недостаточно открытых статистических данных опубликованных прокуратурой либо МВД Российской Федерации. Одним из вариантов получения необходимых данных является дата-майнинг путем парсинга страниц с решениями районных судов. Данный вариант также позволяет выделить ключевые проблемы района в виде наиболее часто встречающихся статей административного и уголовного кодекса Российской Федерации в судебных решениях [3]. На данный момент электронные ресурсы судебных инстанций почти все унифицированы, но при этом результаты судебных заседаний публикуются в максимально сжатом виде и не всегда содержат номер статьи. Далее собранные данные предлагается хранить в реляционной базе данных в нормализованном виде. Слабой стороной предложенного метода является необходимость постоянного парсинга тысяч судебных сайтов, что требует большой вычислительной мощности. На портале правовой статистики пользователи могут загружать xml-файлы содержащие статистические данные, в частности можно получить файл с общим количеством зарегистрированных преступлений в регионах (рис. 2). Полученные из этого файла данные можно использовать как проверочные, таким образом, чтобы числа, указанные в файле, находились как можно ближе к числам полученными в процессе парсинга решения районных судов отдельно взятого субъекта Российской Федерации.

```
<row>
      <value>6835</value>
      <report_period_begin>2017-01-01</report_period_begin>
      <region_name>Пензенская область</region_name>
      <report_period_end>2017-07-01</report_period_end>
</row>
<row>
      <value>10423</value>
      <report_period_begin>2017-01-01</report_period_begin>
      <region_name>Республика Коми</region_name>
      <report_period_end>2017-07-01</report_period_end>
</row>
```

Рис. 2. Фрагмент xml-файла с данными о преступности в регионах

В наше время все больше и больше сервисов «уходит в онлайн», поэтому целесообразнее всего реализовывать картографический сервис отражающий текущее состояние криминального уровня какой-либо конкретной местности, или страны в целом, в виде веб-приложения.

Для выполнения графической интерактивной части существует ряд JS-библиотек, упрощающих разработку. Самые известные из них LeafletJS (http://leafletjs.com/) и Amcharts (https://www.amcharts.com/). В качестве системы менеджмента для статического и новостного контента, а именно для упрощения работы и частичной автоматизации процесса работы с сайтом можно использовать любую из популярных CMS: WordPress, Joomla, Drupal, MODx, Grav и другие. При этом разрабатываемому веб-сервису не потребуется выделенный сервер или дорогой хостинг, достаточно наличия php-сервера версии 5.6 или выше, любой распространенной СУБД и небольшого количества дискового пространства. Перечисленным требованиям удовлетворяют даже самые дешевые предложения на рынке.

Реализуя проект интерактивных карт криминализации районов будет оказана информационная помощь всем гражданам Российской Федерации, имеющим доступ в сеть Интернет. Любой пользователь сможет просматривать и анализировать данные представленные как в текстовом, так и в графическом виде. Предлагаемую концепцию проекта возможно реализовать как отдельный самостоятельный веб-сервис, так и в составе уже существующего инструмента публикации правовой статистики от Генеральной прокуратуры Российской Федерации.

Библиографический список

1. Портал правовой статистики генеральной прокуратуры Российской Федерации. – URL: http://crimestat.ru/
2. Музалева, Т. И. Кластерный анализ в оценке показателей правовой статистики / Т. И. Музалева // Статистика и экономика. – 2012. – № 1. – С. 142–144.
3. Лыгина, Н. И. Исследование правильности и эффективности средств парсинга информации на веб-ресурсах / Н. И. Лыгина, А. Пудич // Инновационная наука. – 2017. – Т. 1, № 3. – С. 59–68.

П. С. Доброжанская

Пензенский государственный университет, Пенза, Россия

АВТОМАТИЗИРОВАННАЯ СИСТЕМА УПРАВЛЕНИЯ ЛОГИСТИКОЙ АВТОТРАНСПОРТНОГО ПРЕДПРИЯТИЯ

Рассматривается подсистема информационной системы автоматизации для управления логистикой автотранспортного предприятия. Степень внедрения – система разработана специально для автотранспортного предприятия. Предполагается, что данная система автоматизирует процессы предприятий, занимающихся логистикой.

Компьютеры и информационные технологии уже давно стали неотъемлемой частью самых разных сфер жизни, и автоматизация управления логистикой автотранспортного предприятия не стала исключением.

Автомобильный транспорт родился на рубеже XIX и XX в., резко изменив при этом жизнь последующих поколений людей. Неслучайно объединение этих двух понятий: автомобильный транспорт и ЭВМ, стали одним из основных направлений развития. Объединение происходит по пути автоматизации собственно процесса перевозки и связанных с ним вспомогательных процессов. Развитие современных средств автоматизации уже достигло тех возможностей, что становится реальным переход от автоматизированных систем управления транспортом к автоматическим, при котором будет устранена необходимость присутствия человека в кабине автомобиля.

Широко автоматизация проникла и в сферу управления автотранспортными предприятиями. Крупные фирмы по созданию информационно-управленческих систем разрабатывают специально для автопредприятий программные комплексы, направленные на повышение эффективности бизнес-процессов путем новой организации их экономической деятельности.

Информация в современном мире превратилась в один из наиболее важных ресурсов, а ИС (информационные системы) стали необходимым инструментом во всех сферах деятельности.

Верно это и для автотранспортного предприятия. Информационные процессы присутствуют во всех областях данной отрасли. От их упорядоченности зависит четкость функционирования системы в целом и эффективность управления ею, поэтому ключевое звено в данной работе является информационная система.

Внедрение таких информационных систем способны значительно усовершенствовать рабочие процессы на самых разных уровнях: начиная от создания автоматизированного рабочего места до комплексной автоматизации предприятия.

Сегодня, ведение документации отнимает у предпринимателя слишком много времени. Автоматизация предприятия направлена на составление договоров, создания различ-

ных протоколов, выписок, регистрационных данных и пр. Предлагаемое программное обеспечение для автотранспортного предприятия позволяет легко вести полный учет всех оказанных услуг.

Наиболее актуальной решаемой задачей является эксплуатации автомобильного транспорта с использованием систем управления, систем передачи данных и систем мониторинга транспортных средств. В частности, таким инструментом развития является автоматизированная система транспортной логистики. Как правило, системы автоматизации грузоперевозок дают возможность решать такие задачи, как накопление и представление в удобном для анализа виде фактических данных об использовании транспорта, обеспечение ежедневного контроля над отклонениями фактических параметров использования автотранспорта от запланированных [1].

Первые такие информационные системы появились на свет достаточно давно. На современном российском рынке можно выделить ряд таких коммерческих информационных систем, таких как «TransTrade», «TransBase», «Грузоперевозки 2.0» и многие другие, как менее известные, так и широкомасштабное программное обеспечение.

В настоящее время аналогов таких систем на рынке программного обеспечения существует не очень большое количество. Помимо крупных информационных систем, затрагивающих все сферы деятельности, существуют узкоспециализированные системы. Такие системы нацелены на регулирования деятельности в соответствии с его профилем и отражают специфичную для отрасли модель предметной области [2].

Недостатками крупных информационных систем является их громоздкость. Для обслуживания таких систем должны быть специалисты высокого уровня по администрированию, как самой информационной системы, так и сопутствующих ей продуктов (операционная система, СУБД). Стоимость крупных информационных систем и сопутствующих продуктов, как правило, слишком велика для небольших автотранспортных предприятий, например, таких как автотранспортные предприятия, занимающиеся доставкой грузов по России. Некоторые частные предприятия так же не всегда могут позволить себе приобрести полнофункциональную информационную систему. Поэтому в основу данного проекта легла идея создания компактной, удобной в использовании информационной системы автоматизации деятельности автотранспортного предприятия, функционал которой должен удовлетворять обобщенным потребностям, без учета его конкретной специфики.

Как правило, информационная система автоматизации управления логистикой автотранспортного предприятия включает в себя следующие подсистемы:

- подсистема управления экономической деятельностью [3];
- подсистемы управления деятельностью данному предприятию [4].

Целиком задача создания информационной системы достаточно трудоемка.

Цель данного проекта является создание части информационной системы, обеспечивающей некоторый функционал.

Данная система должна включать следующие функции:

- Добавление, удаление данных.
- Поиск. Данная функция может выполняться по искомому ключу.
- Создание договоров, различных экономических отчетов.
- Дополнительно в рамках создания подсистемы реализации подлежит базовая серверная часть системы, обеспечивающая выполнение общих для всех подсистем функций.

В результате разработки данного проекта должны будут спроектированы и реализованы на методологическом, алгоритмическом и программном уровне компоненты информационной автоматизированной системы для управления логистикой автотранспортного предприятия.

В итоге должна быть разработана концепция информационной системы автоматизации управления логистикой автотранспортного предприятия, позволяющая вводить и анализировать информацию логистических данных, а так же сгенерировать отчетную деятельность предприятия.

Таким образом, следует отметить, что автоматизированная система, разработанная или доработанная в соответствии с проектом, позволит с минимальными материальными и временными издержками решать поставленные задачи.

Библиографический список

1. Организация производства и управления предприятием : учеб. / под ред. О. Г. Туровца. – М. : ИНФРА-М, 2006. – 326 с.
2. Петрова, Е. В. Статистика транспорта : учеб. / Е. В. Петрова ; под ред. М. Р. Ефимовой. – М. : Финансы и статистика, 2008. – 366 с.
3. Организация, планирование и управление в автотранспортных предприятиях / под ред. М. П. Улицкого. – М. : Транспорт, 2010. – 346 с.
4. Бычков, В. П. Экономика автотранспортного предприятия : учеб. / В. П. Бычков. – М. : ИНФРА-М, 2009. – 384 с.

Я. Г. Курденкова, И. П. Бурукина

Пензенский государственный университет, Пенза, Россия

РАЗРАБОТКА СИСТЕМЫ ПОДДЕРЖКИ ПРОВЕДЕНИЯ КОНФЕРЕНЦИИ

Разработка системы, предназначенной для повышения эффективности проведения конференции и снижение трудовых затрат организаторов за счет автоматизации сбора, учета и обработки данных.

Несмотря на высокую степень развития компьютерных технологий и широкое распространение автоматизированных компьютерных средств в различные области деятельности человека, в большинстве случаев, работа по организации, ведению учета и информированию участников выполняется членами комиссии конференции вручную, что крайне неэффективно и может повлечь за собой немалое количество неточностей и проблем [1].

В общем случае, процесс организации и проведения конференции можно разделить на три этапа:

- сбор заявок на участие в конференции;
- рассмотрение и утверждение заявок членами комиссии;
- проведение секционных заседаний.

В настоящее время, в большинстве случаев, для участия в конференции в качестве заявки необходимо отправлять свои данные и текст работы на электронный почтовый ящик секретариата конференции.

При таком способе организации данных, их обработка является трудоемкой для члена комиссии, обрабатывающего поступающие заявки вручную, а также не обеспечивается даже начальный уровень безопасности контактных данных.

Использование системы поддержки проведения конференции может поспособствовать снижению трудовых затрат организаторов и уменьшению вероятности возникновения ошибок в процессе работы.

В процессе анализа предметной области, обзора возможных аналогов, таких как Advanced Mass Sender Enterprise, Unisender, MailChimp, YConfs [2], а также основополагающих задач, связанных с организацией вузовской научно-практической конференции, была выявлена потребность в создании системы, способной частично или полностью автоматизировать работу секретаря конференции и отвечающей следующим основным требованиям к функциональности:

- автоматизация массовых почтовых рассылок;

- формализованный сбор персональных данных;
- обработка полученных данных, анализ данных;
- формирование отчета.

Выбор средств разработки реализуемой системы был сделан исходя из необходимости обеспечения следующих характеристик:

- кроссплатформенность системы;
- доступность с различного вида устройств;
- отсутствие привязки к одному рабочему месту;
- относительная простота поддержки и модификации.

«Каркас» системы был создан на стандартизированном языке разметки документов HTML с использованием каскадных таблиц стилей CSS [3]. «Динамика» системы была создана при помощи языков программирования JavaScript[4] и PHP [5]. В качестве среды разработки был выбран Chrome Developer Tools, представляющий собой набор набор инструментов для веб-разработки и отладки, встроенный в браузер Google Chrome. Для хранения данных используется MySQL – компактный многопоточный сервер баз данных, поддерживающий язык запросов SQL [6]. Работа по разработке системы была развернула на локальном сервере, в качестве инструментария для обеспечения функционирования которого был использован базовый пакет Denwer [7].

Разработанная система предназначена для повышения эффективности проведения конференции за счет автоматизации сбора, учета и обработки данных, помощи в планировании работы, автоматической рассылки информационных почтовых уведомлений, на основе данных из таблиц базы данных.

Работа системы нацелена на взаимодействие с двумя основными независимыми друг от друга группами пользователей, к которым относятся:

- потенциальный участник конференции, желающий подать заявку на участие;
- администратор системы – секретарь комиссии конференции, отвечающий за работу с заявками;

Поэтому, было принято решение разделить структуру разрабатываемой системы на две составляющие:

- часть системы, ориентированная на работу с заявителями – форма заявки на информационной странице конференции;
- административная часть системы, функционал которой позволит обеспечить снижение трудозатрат секретаря конференции на учет и обработку данных;

В ходе разработки системы были реализованы следующие функции:

- вывод страницы с формой подачи заявок на участие и управление ей непосредственно из административной части системы;
- организация сбора заявок на участие в виде набора персональных данных и текстового файла с научной работы;
- создание административной части системы с разграничением прав доступа;
- отображение списка всех полученных заявок с сокрытием персональных контактных данных.
- возможность редактирования и удаления заявок;
- возможность редактирования организационной информации.
- организация массовых почтовых рассылок информационных писем с автоматически генерируемым из базы данных содержимым;
- формирование отчета об итогах проведения конференции;
- формирование статистических графиков и диаграмм по данным, сформированным из запросов к базе данных.

Работа системы нацелена на достижение снижения трудозатрат потенциального пользователя административной части, поэтому данная часть системы разрабатывалась таким образом, чтобы взаимодействие с элементами меню и страниц было как можно более простым для восприятия и интуитивно понятным для рядового работника без специального уровня технических знаний.

Таким образом, в результате работы была спроектирована и реализована система поддержки проведения конференции, отвечающая основным требованиям, выявленным на этапе анализа предметной области и выполняющая заявленные для реализации функции. Разработанная система может использоваться в учебных заведениях, учреждениях и организациях, где проводятся научные конференции.

Библиографический список

1. Бурукина, И. П. Использование инструментов интернет-маркетинга при организации конференций / И. П. Бурукина, Я. Г. Курденкова, А. И. Кириллов // Университетское образование (МКУО–2016) : сб. ст. – Пенза, 2016. – С. 90–92.

2. Сибирский федеральный университет: о системе Yconfs. – URL: http://yconfs.sfu-kras.ru/ (дата обращения: 13.02.2017).

3. Бурукина, И. П. Возможности платформы HTML5 / И. П. Бурукина, Я. Г. Курденкова // Университетское образование (МКУО–2016) : сб. ст. – Пенза, 2015. – С. 214–216.

4. Javascript.ru: Современный учебник Javascript. – URL: https://learn.javascript.ru/ (дата обращения: 02.04.2017).

5. Руководство по PHP. – URL: https://secure.php.net/manual/ru/ (дата обращения: 20.03.2017).

6. Веллинг, Л. Разработка веб-приложений с помощью PHP и MySQL / Л. Веллинг, Л. Томсон. – 4-е изд. – М. : Вильямс, 2010.

7. Denwer: Денвер – локальный сервер. – URL: http://www.denwer.ru/ (дата обращения: 20.03.2017).

А. В. Лисенков, А. С. Гюлзатян, Е. А. Поменкова

ОАО «НПП «Рубин», Пенза, Россия

ОСНОВНЫЕ ПРЕИМУЩЕСТВА ВЕДЕНИЯ ЭЛЕКТРОННОГО ФОРМУЛЯРА В ИНТЕРЕСАХ АВТОМАТИЗИРОВАННОГО УПРАВЛЕНИЯ ЖИЗНЕННЫМ ЦИКЛОМ ИЗДЕЛИЙ

Приводятся основные положения по ведению электронного формуляра в интересах автоматизированного управления жизненным циклом изделий.

Одним из важнейших документов, используемых в течение длительного жизненного цикла изделия (технического средства) и позволяющего фиксировать события, произошедшие с ним, а также значения технических характеристик изделия и его основного оборудования (составных частей) является формуляр (технический паспорт) изделия.

Однако, количество различного рода изделий, их характеристик и прочей информации, фиксируемой в процессе эксплуатации, огромно, а операции с данными необходимо проводить в кратчайшие сроки. Не следует забывать о полноте, достоверности (актуальности) и защите контролируемой информации, а также важности ее оперативного получения в режиме реального времени.

Ручное заполнение бумажных (паспорт, формуляр) или электронных версий документов на изделие, ведение не взаимодействующих между собой баз данных по отдельным типам изделий даже в совокупности не дают возможность сформировать единую модель информационной среды, решающей задачи контроля состояния как отдельно взятого образца технического средства (ТС), так и мониторинга обеспеченности (укомплектованности) потребителя техникой в целом [1]. Одновременно существуют проблемы ведения детальной информации о результатах эксплуатации изделий; учета запасных инструментов и принадлежностей; контроля объемов имеющихся запасов [2].

Одним из решений данной проблемы является применение автоматизированной системы (АС) мониторинга жизненного цикла изделия и состояния обеспеченности потребителя, важным элементом которой является инструмент ведения учетных данных и формирования документов о результатах эксплуатации изделия – унифицированный электронный формуляр (ЭФО).

ЭФО объединяет в себе следующие технические и эксплуатационные данные об изделии, фиксируемые на протяжении его жизненного цикла:

– основные сведения (идентификационные данные) об изделии и предприятии-изготовителе (разработчике);
– основные технические данные;
– комплектность;
– ресурсы, сроки службы и хранения, гарантия изготовителя;
– консервация, упаковывание, хранение и приемка изделия;
– движение изделия при эксплуатации;
– учет работ (выполнение работ) при эксплуатации;
– учет технического обслуживания и ремонт изделия;
– сведения об утилизации;
– контроль состояния изделия и ведения формуляра [3].

На рис. 1 представлен внешний вид основных диалоговых форм ЭФО.

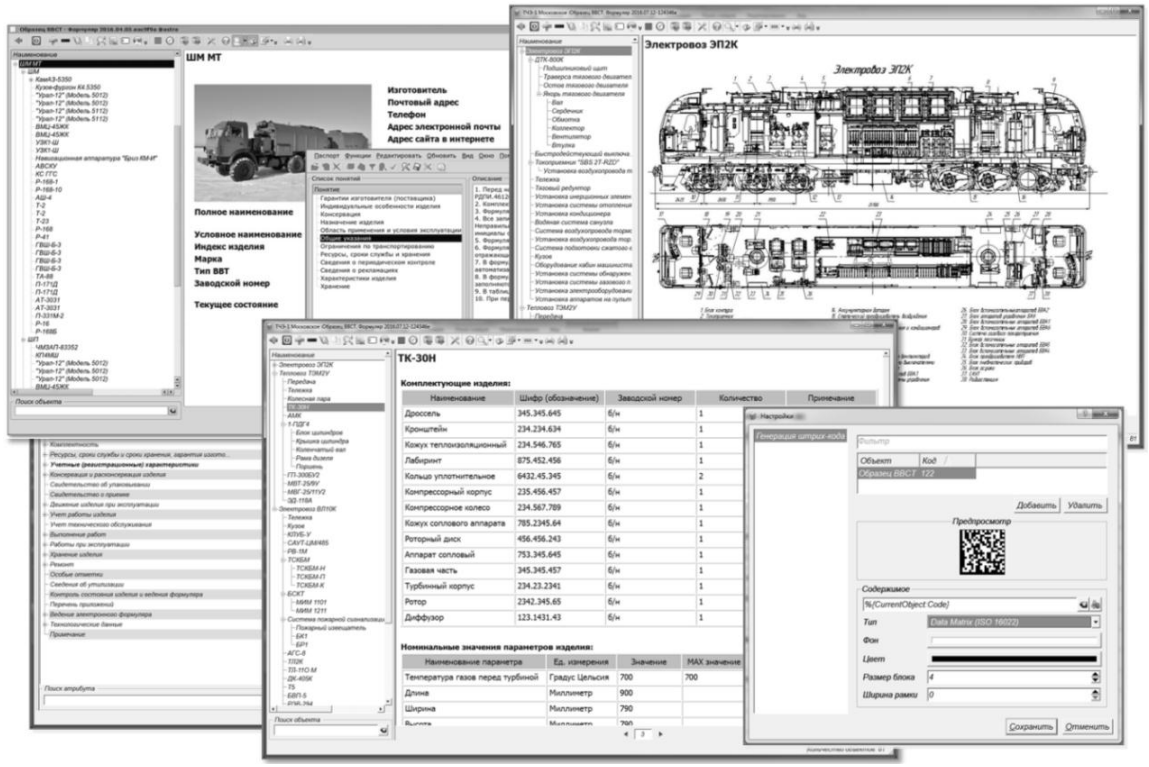


Рис. 1. Внешний вид основных диалоговых форм ЭФО (данные условные)

Унификация ЭФО достигается благодаря единой модели информационной среды с возможностью расширения описания информационных объектов по мере увеличения знаний или изменений предметной области, фиксированного набора требований и правил к ведению (заполнению) ЭФО и сопутствующей нормативно-справочной информации, способам сбора и передачи данных. А также за счет реализации требований эксплуатационной документации и ГОСТ, предъявляющих особые требования к построению, изложению, оформлению и содержанию паспортов и формуляров (например: авиационная техника).

Функционал встроенных в АС средств автоматической идентификации предусматривает формирование уникального двумерного штрих-кода (в перспективе возможно применение RFID-меток) для последующего маркирования ТС и его основного оборудования, поиск по штрих-коду и идентификация изделий в базе данных ЭФО для просмотра и обработки данных.

Возможность формирования штриховых кодов особенно актуальна на первоначальном этапе внедрения АС и позволяет сопоставить информацию в бумажных версиях документов и ЭФО, что обеспечивает выявление следующих возникающих проблем:

– расхождение сведений о фактически установленном в изделии оборудовании и указанным в документации;

– отсутствие знаков идентификации на некоторых изделиях (нечитаемые заводские номера, клейма, отсутствие табличек изготовителей и т.п.);

– наличие изделий с отсутствующими бумажными формулярами, паспортами;

– наличие паспортов-дубликатов, в которых присутствует различная информация об изделии [4].

АС мониторинга жизненного цикла изделия и состояния обеспеченности позволяет организовать информационное взаимодействие между участниками жизненного цикла изделия – разработчиками, предприятиями-изготовителями, потребителями и ремонтными органами. Это особенно актуально для организаций, где парк и номенклатура ТС достаточно широки, а вопросы, касающиеся эксплуатации техники возникают на всех уровнях управления (Минобороны, МВД, МЧС, РЖД и др.).

Преимущества АС мониторинга жизненного цикла изделия и состояния обеспеченности выражаются в следующем:

– способность осуществления оперативного мониторинга состояния изделия и его основного оборудования (составных частей);

– определение потребности в запчастях и комплектующих;

– оценка возрастной структуры, отслеживание стоимости эксплуатации и обслуживания изделия;

– анализ информации по надежности, отказам с учетом сроков эксплуатации, а также определение степени деградации параметров изделия в процессе эксплуатации (неспособности в полном объеме выполнять требуемую функцию);

– автоматизация статистической отчетности работы изделия (формирование отчетной документации, сводных таблиц, диаграмм и графиков);

– анализ состояния «аварийного» оборудования (требующего особого контроля).

На рис. 2 представлена диаграмма сравнения абстрактной величины (показателя) затрат ресурсов (финансовых, материальных, производственных) в зависимости от времени эксплуатации и возникающих отказах (сбоях в работе) изделия.

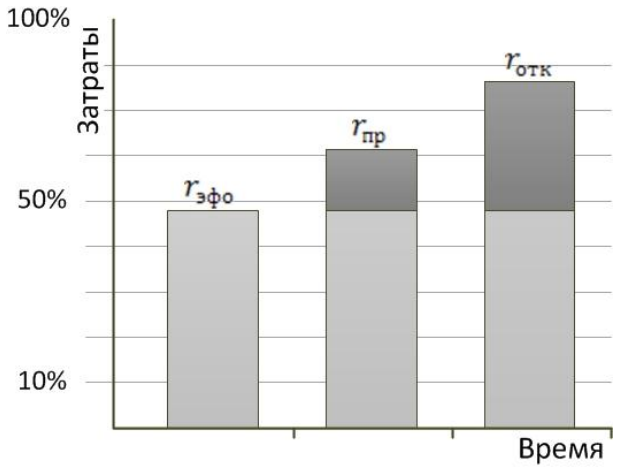


Рис. 2. Диаграмма сравнения показателей затрат ресурсов (вариант)

На диаграмме моменты отказа изделия или оборудования, входящего в его состав, можно представить как:

$r_{отк}$ – затраты, возникающие на момент фактического отказа изделия;

$r_{пр}$ – затраты, возникающие на момент потенциального отказа изделия (проведение регламентных работ согласно рекомендациям разработчика, касающихся технического обслуживания, ремонта и т.д.);

$r_{эфо}$ – затраты, возникающие на момент прогнозируемого отказа изделия и полученные в результате автоматизированного анализа данных ЭФО.

Из графика видно следующее:

$$r_{эфо} < r_{пр} < r_{отк}.$$

Это, в свою очередь, подтверждает целесообразность проведения дальнейших исследований в части решения задач оптимизации структуры ремонтного цикла и затрат на восстановление (эксплуатацию), технического обслуживания и ремонта изделий, проведения анализа состояния ТС и оборудования в интересах автоматизированного управления жизненным циклом изделий.

Библиографический список

1. Лисенков, А. В. Информационные технологии в решении задач контроля состояния обеспеченности Вооруженных сил Российской Федерации вооружением, военной и специальной техникой / А. В. Лисенков, С. А. Медовщикова // Вопросы радиоэлектроники. Сер.: СОИУ. – 2016. – № 12. – Вып. 1. – С. 26–30.

2. Буренок, В. М. Проблемы создания системы управления полным жизненным циклом вооружения, военной и специальной техники / В. М. Буренок // Вооружение и экономика. – 2014. – № 2 (27) . – С. 4–9. – URL: http://www.viek.ru.

3. ГОСТ 2.612–2011 Единая система конструкторской документации. Электронный формуляр. Общие положения. – М. : Стандартинформ, 2012. – 23 с.

4. Электронный паспорт локомотива: преимущества и перспективы // Локомотив. – 2012. – № 10 (670). – С. 6–7.

Д. С. Мазилкин

Пензенский государственный университет, Пенза, Россия

СПОСОБЫ ОБРАБОТКИ WEB-СТРАНИЦ НА PHP

Описаны преимущества и недостатки наиболее часто используемых способов обработки html кода на PHP.

Довольно часто веб-разработчику нужно получить информацию со сторонних сайтов. Это необходимо для заполнения своего сайта различным контентом или для динамической загрузки, какой-либо информации. Существует множество библиотек предназначенных для получения и обработки html кода. Перед разработчиком возникает вопрос: какой способ парсинга выбрать? В этой статье описаны преимущества и недостатки некоторых из них.

Ниже перечислены самые популярные возможности обработки html кода:

- регулярные выражения;
- DOM и XPath;
- Simple HTML DOM;
- phpQuery;

- htmlSQL;
- Zend_DOM_Query.

Регулярные выражения – это простой и популярный способ парсинга, так как они работают на порядок быстрее, чем любые библиотеки. Однако их сложно использовать для обработки большого и сложного HTML-кода. Приходится выполнять обработку в несколько этапов сначала выделяя большие части, затем разбивая их на более мелкие. Так же они не подходят для обработки html кода, если он содержит синтаксические ошибки или не соответствует какому-то определенному шаблону.

Начиная с пятой версии, в PHP встроены DOM и XPath. Они достаточно быстры и просты в освоении. Однако движок, используемый для парсинга, изначально предназначен для обработки XML. И хотя XML и HTML во многом схожи, у них есть и различия. Из-за этого возникают некоторые ограничения по обработке html. Например, необходимо, что бы все теги были закрыты. Их основное преимущество в том, что они уже встроены в PHP, что позволяет не искать сторонние решения.

Simple HTML DOM – PHP-библиотека, которая позволяет обрабатывать HTML-код при помощи jQuery-подобных селекторов. Она очень удобна и достаточно проста в освоении. Так же библиотека способна обрабатывать даже невалидный html код, это делает ее еще более комфортной в использовании. При помощи данной библиотеки можно обращаться к элементам и атрибутам, искать вложенные элементы определенного уровня, фильтровать их, а так же искать текст и комментарии. Еще одним плюсом является отсутствие проблем с кодировками, преобразование происходит автоматически.

Единственным значительным минусом библиотеки является низкая скорость работы. По этому параметру она проигрывает всем своим конкурентам. Впрочем, этот недостаток имеет значение только при обработке больших объемов информации, так как получение html занимает на порядок больше времени, чем его обработка.

Так же стоит упомянуть, что среди аналогов библиотека наиболее распространена в русскоязычном интернете. Что значительно упрощает работу с ней для новичков и людей, плохо владеющих английским языком.

PhpQuery это php вариант известного javascript фраемворка – jQuery, с точно таким же синтаксисом. В сравнении с Simple HTML DOM библиотека еще более схожа с оригиналом. Автор портировал практически все, что есть в jQuery: селекторы, события, обход, манипуляции, AJAX, атрибуты и плагины. Все это позволяет легко перемещаться по DOM дереву html документа и с легкостью находить нужные элементы. Обладая практически всеми преимуществами Simple HTML DOM, она значительно быстрее работает. Однако при использовании могут возникнуть проблемы с кодировками.

HtmlSQL – экспериментальная PHP библиотека, которая дает возможность работать с HTML при помощи синтаксиса SQL. Методы fetch_array() и fetch_objects() так же как в mysql позволяют получить результат запроса в виде ассоциативного массива или объекта. HtmlSQL обладает хорошим быстродействием, зачастую работая быстрее phpQuery. Однако библиотеке не хватает функционала для решения сложных задач, и ее разработка прекращена еще в 2012 г.

Zend_DOM_Query (ZDQ) – предоставляет возможности для работы с XML и (X)HTML документам при помощи селекторов XPath и CSS. ZDQ полностью интегрирована для Zend Framework, однако есть возможность использования вне данного фреймворка. Библиотека обладает высокой скоростью работы, особенно это заметно с большими объемами данных. К сожалению, официальная документация библиотеки оставляет желать лучшего. Если же не используется Zend Framework, для ее функционирования нужно, что бы пять разных файлов были размещены в определенных папках относительно корневого каталога, что не совсем удобно.

Есть еще множество похожих библиотек, однако они не так часто используются, обладают не таки широким набором возможностей или значительно менее производительны, чем представленные.

Ниже приведена таблица, в которой в более наглядной форме показаны преимущества и недостатки всех описанных методов обработки.

Таблица

Сравнение способов обработки html

Способ обработки	Быстродействие	Функционал	Прост в освоении
Regular expressions	+	–	–
XPath	+	–	+
Simple HTML DOM	–	+	+
PhpQuery	+	+	+
HtmlSQL	+	–	+
Zend_DOM_Query	+	+	–

В конечном итоге лучшей по совокупности характеристик является библиотека XPath. Она достаточно быстро работает, обладает большим набором возможностей, проста в освоении и использовании. Однако выбирать ее не всегда целесообразно. Для простых задач рациональнее использовать регулярные выражения или XPath. Для новичков же, за счет простоты и большей распространенности, неплохим выбором станет Simple HTML DOM.

Библиографический список

1. URL: https://phpdoc.hotexamples.com/class/-/phpQuery

2. URL: http://php.net/manual/ru/class.domxpath.php

Ю. В. Давыдова

Орловский государственный университет им. И. С. Тургенева, Орел, Россия

ПРОБЛЕМА ОБРАБОТКИ ОШИБОК В ТЕКСТАХ СООБЩЕНИЙ ПОЛЬЗОВАТЕЛЕЙ В ЗАДАЧЕ МОНИТОРИНГА ВИРТУАЛЬНЫХ СОЦИАЛЬНЫХ СЕТЕЙ

Рассмотрены ошибки в текстовых сообщениях пользователей и, как следствие, возникающие сложности при мониторинге виртуальных социальных сетей. Приведен обзор методов коррекции ошибок. Сделан вывод о необходимости интегрирования модели ошибок в нечеткий текстовый поиск, осуществляемый в процессе мониторинга.

Мониторинг виртуальных социальных сетей (ВСС) представляет собой систематический сбор текстовых сообщений, информации о пользователях и последующий поиск по набору ключевых слов, характеризующих искомую тематику [1]. В случае простого поиска на совпадение с ключевыми словами некоторое множество релевантных результатов не будет выявлено, поскольку текстовые сообщения содержат ошибки.

Ошибки можно разделить на орфографические и типографические. Следует отметить, что для некоторых ошибок такая классификация не всегда является однозначной. Согласно Дамерау [2] 80 % всех ошибок представляют собой единичную ошибку, принадлежащую к одному из четырех классов ошибок: пропуск буквы («сонце» вместо «солнце»), замена буквы («симпотичный» вместо «симпатичный»), вставка буквы («парраллельный» вместо «параллельный») и перестановка двух соседних букв («кокши» вместо «кошки»).

В целом текстовые сообщения в социальных сетях можно охарактеризовать следующим образом [3, 4]:

– использование разговорного стиля при письме, использование сокращений, аббревиатур, иноязычных заимствований и сленга;

– синтаксические конструкции могут носить фрагментированный характер, формальные синтаксические связи ослабляются, текст становится прерывистым;

– небольшой размер сообщений в среднем.

Таким образом, необходимо использовать дополнительные методы, позволяющие сделать поиск менее чувствительным к ошибкам. Рассмотрим подходы к исправлению ошибок в текстах.

Техники коррекции ошибок можно классифицировать по разным признакам. В данной работе рассмотрим учет контекста, при этом подходы к исправлению ошибок подразделяются на [5, 6]:

– методы коррекции изолированного термина, при этом осуществляется исправление каждого отдельного слова без учета контекста;

– методы коррекции с учетом контекста.

Первые системы коррекции ошибок реализовывали только методы коррекции изолированного термина. При формировании списка кандидатов на коррекцию для ошибочно написанного слова перебирались все допустимые с учетом единичного редакционного расстояния комбинации вставки, удаления и перестановки букв [2]. Редакционное расстояние является мерой близости между двумя словами, оно представляет собой количество операций редактирования, необходимых для превращения одного слова (в общем случае строки символов) в другое [6, 7]. В зависимости от типа разрешенных операций выделяют различные редакционные расстояния, одним из часто применяемых является расстояние Левенштейна, при котором разрешены операции вставки, удаления и замены символа. Если дополнительно разрешена операция транспозиции, то такое расстояние является расстоянием Дамерау-Левенштейна. Вместо полного перебора возможных вариантов изменения слова с ошибкой с учетом допустимых операций редактирования, для нахождения кандидатов на коррекцию стали применять расстояния Левенштейна или Дамерау-Левенштейна. В настоящее время системы коррекции умеют ранжировать кандидатов и автоматически исправлять ошибки.

Одним из подходов к исправлению ошибок является метод n-грамм. Например, в работе [8] после детектирования ошибки с использованием словаря, авторы выявляют кандидатов на коррекцию на основе количества общих биграмм, при этом учитывается позиция биграмм в слове. Биграмма в позиции n в ошибочно написанном слове сравнивается с биграммами в позициях $n - 1$, n, $n + 1$ в слове-кандидате. При поиске кандидатов на коррекцию авторы исходят из предположения, что ошибка в первой букве слова крайне маловероятна.

На основе n-грамм для коррекции ошибок была разработана модель зашумленного канала (в контексте письменной речи зашумленный канал – это человек, печатающий текст) [9]. Рассматриваемый подход использует теорему Байеса и заключается в следующем: пусть x – ошибочно написанное слово, из множества W возможных кандидатов на коррекцию необходимо выбрать лучший вариант, такой, что

$$w = \arg\max_{w \in W} P(w \mid x) = \arg\max_{w \in W} \frac{P(x \mid w)P(w)}{P(x)}.$$

Таким образом, для определения лучшего слова необходимо максимизировать значение $P(x|w)P(w)$, где $P(x|w)$ – модель зашумленного канала (или модель ошибок), представляющая собой вероятность того, что слово w может быть искажено в x, а $P(w)$ – языковая модель (словарная модель вероятности). Условные вероятности подсчитываются на основе текстового корпуса для каждого типа ошибки – вставки, удаления, замены и транспозиции. При этом формируются матрицы, строками и столбцами которых являются буквы алфавита, пересечение которых представляет собой количество раз, когда та или иная ошибка встречалась в корпусе для соответствующих пар букв.

Рассмотренный подход на основе модели зашумленного канала был улучшен в [10]. В модели ошибок при вычислении условных вероятностей учитываются операции редактирования не только для одиночных букв, но и группы букв (подстрок). В работе [11] в модель ошибок отдельным компонентом интегрирована фонетическая модель. Авторы выявляют вероятности различных вариантов произношения той или иной буквы и формулируют правила произношения с учетом расположенных рядом букв. При формировании кандидатов на коррекцию слова переводятся в код согласно фонетической модели и затем сравниваются с кодом ошибочно написанного слова. Следует отметить, что если языковая модель в подходе зашумленного канала представлена униграммной моделью, то подход является методом коррекции изолированного термина. Если языковая модель – модель биграмм, триграмм и т.д., то подход будет являться методом исправления ошибок с учетом котекста.

Существуют методы коррекции ошибок на основе строгих морфологических правил, например [12]. При этом слово рассматривается как совокупность морфем, подлежащих проверке и исправлению по отдельности. Данный подход не получил широкого распространения.

Еще одной ветвью методов исправления ошибок являются методы, использующие ключи, сгенерированные по определенным правилам для слов, как, например, в [13]. Правила являются некоторыми эвристиками, например, удаление из слова малозначащих букв, замена двух согласных одной буквой (в случае их удвоения) и др. При выявлении списка кандидатов сравниваются не сами слова, а их ключи. Данная техника также не получила достаточного распространения.

Для учета контекста используются модели n-грамм на словах. Например, в [14] для коррекции ошибок применяется модель зашумленного канала, но языковая модель $P(w)$ аппроксимируется триграммной моделью (в отличие, например, от [8], где рассматриваются униграммы), а условная вероятность $P(x|w)$ рассчитывается для триграмм.

В работе [15] исправление ошибок осуществляется на основе набора Google n-gram, при этом для детектирования ошибок в качестве словаря выступает коллекция униграмм на словах. Ошибочно написанное слово разбивается на буквенные биграммы, затем осуществляется их поиск в словах из набора униграмм. В результате поиска слова, содержащие наибольшее количество буквенных биграмм, формируют множество кандидатов на коррекцию. Для выбора лучшего кандидата применяются 5-граммы на словах, при этом выбирается левый контекст от слова с ошибкой, вместо ошибочного слова подставляется кандидат на коррекцию, и все выражение ищется в словаре 5-грамм. Следует отметить, что для русского языка не существует столь обширных лингвистических ресурсов.

Среди всех техник исправления ошибок в письменной речи наибольшее распространение получил подход на основе модели канала с ошибками. Данный метод был усовершенствован и применяется для коррекции запросов пользователей в современных поисковых системах таких, как Яндекс и Рамблер [16, 17].

Основная задача мониторинга ВСС – эффективный поиск текстовых сообщений по ключевым словам выбранной тематики. Возможность находить соответствия в тексте с задаваемой мерой близости предоставляют методы нечеткого поиска [6, 7]. Однако результат их работы может быть неудовлетворительным, особенно при поиске коротких слов. Например, при условии возможности единичной ошибки в результирующую выборку попадут слова «рама» и «мама», «карова» и «корова». Первый результат является, несомнен-

но, ошибочным. Для улучшения нечеткого поиска следует использовать подходы, применяемые для коррекции ошибок. При этом нет необходимости формировать множество кандидатов на коррекцию и выбирать лучший вариант. Вследствие этого, предлагается использовать только модель ошибок из модели зашумленного канала [9]. Модель ошибок аккумулирует в себе знания о языке и о некотором возможном поведении пользователя при наборе текста. Кроме того, предлагается дополнить ее списком возможных клавиатурных опечаток, а также ввести в модель знания о наиболее распространенных фонетических ошибках русского языка.

Библиографический список

1. Савва, Ю. Б. Мониторинг и анализ сообщений участников виртуальных социальных сетей как средство обеспечения информационно-психологической безопасности личности / Ю. Б. Савва, Ю. В. Давыдова, В. Т. Еременко // Информационная безопасность и защита персональных данных. Проблемы и пути их решения : материалы VIII Всерос. науч.-практ. конф. – Брянск : Изд-во БГТУ, 2016. – С. 129–136.

2. Damerau, F. J. A technique for computer detection and correction of spelling errors / F. J. Damerau // Communications of the ACM. – 1964. – Vol. 7, № 3. – P. 171–176.

3. Валгина, Н. С. Активные процессы в современном русском языке : учеб. пособие для студентов вузов / Н. С. Валгина. – М. : Логос, 2003. – С. 304.

4. Савва, Ю. Б. О проблеме лингвистического анализа сленга в задаче автоматизированного поиска угроз распространения наркомании в виртуальных социальных сетях / Ю. Б. Савва, В. Т. Еременко, Ю. В. Давыдова // Информационные системы и технологии. – 2015. – № 6 (92). – С. 68–75.

5. Kukich, K. Techniques for automatically correcting words in text / K. Kukich // ACM Computing Surveys. – 1992. – Vol. 24, № 4. – P. 377–439.

6. Manning, Ch. D. Introduction to information retrieval / Ch. D. Manning, P. Raghavan, H. Schutze. – Cambridge University Press, 2008. – P. 496.

7. Navarro, G. A guided tour to approximate string matching / G. Navarro // ACM Surveys. – 2001. – Vol. 33, № 1. – P. 33–88.

8. Ahmed, F. Revised N-Gram based automatic spelling correction tool to improve retrieval effectiveness / F. Ahmed, E. W. De Luca, A. Nurnbenger // Polibits. – 2009. – Vol. 40. – P. 39–48.

9. Kernighan, M. D. A spelling correction program based on a noisy channel model / M. D. Rernighan, K. W. Church, W. A. Gale // Proceedings of COLING. – 1990. – P. 205–210.

10. Brill, E. An improved error model for noisy channel spelling correction / E. Brill, R. C. Moore // Proceedings of the 38th Annual Meeting on the Association for Computational Linguistics. – 2000. – P. 286–293.

11. Toutanova, K. Pronunciation modeling for improved spelling correction / K. Toutanova, R. C. Moore // Proceedings of the 40th Annual Meeting on Association for Computational Linguistics. – 2002. – P. 144–151.

12. Spelling corrector for Basque based on morphology / I. Aduriz, M. Urkia, I. Alegria, X. Artola, N. Ezeiza, K. Sarasola // Literary and Linguistic Computing. – 1997. – Vol. 12, № 1. – P. 31–38.

13. Pollock, J. J. Automatic spelling correction in scientific and scholarly texts / J. J. Pollock, A. Zamora // Communications of the ACM. – 1984. – Vol. 27, № 4. – P. 358–368.

14. Mays, E. Context based spelling correction / E. Mays, F. J. Damerau, R. L. Mercer // Information processing and management. – 1991. – Vol. 27, № 5. – P. 517–522.

15. Bassil, Y. Context-sensitive spelling correction using Google Web 1T 5-Gram information / Y. Bassil, M. Alwani // Computer and Information Science. – 2012. – Vol. 5, № 3. – P. 37–48.

16. Карпенко, М. П. Некоторые методы очистки словаря запросов поиска / М. П. Карпенко, С. В. Протасов // Компьютерная лингвистика и интеллектуальные технологии : материалы ежегодной Междунар. конф. «Диалог». – М. : Изд-во РГГУ, 2011. – С. 280–291.

17. Панина, М. Ф. Автоматическое исправление опечаток в поисковых запросаз без учета контекста / М. Ф. Панина, А. В. Байтин, И. Е. Галинская // Компьютерная лингвистика и интеллектуальные технологии : материалы ежегодной Междунар. конф. «Диалог». – М. : Изд-во РГГУ, 2013. – С. 568–579.

А. М. Джамбеков

Астраханский государственный технический университет, Астрахань, Россия

ПРИНЯТИЕ ПАРЕТО-ОПТИМАЛЬНЫХ РЕШЕНИЙ ПРИ УПРАВЛЕНИИ ПРОЦЕССОМ КАТАЛИТИЧЕСКОГО РИФОРМИНГА

Введен обобщенный критерий оптимальности управления процессом каталитического риформинга. Построено двумерное критериальное пространство для принятия Парето-оптимальных решений при управлении процессом каталитического риформинга. Из множества неулучшаемых решений определено оптимальное по Парето решение при управлении процессом каталитического риформинга.

Каталитический риформинг (КР) является важнейшим технологическим процессом современной нефтепереработки и нефтехимии. Управление процессом КР осуществляется с использованием различных критериев оптимальности (производительность, затраты на производство бензина, октановое число бензина и пр.) [1].

Для производства бензина важно не только повышение его октанового числа *ON*, но и снижение затрат на его производство *Z*. Поэтому необходимо объединение данных требований в обобщенный критерий оптимальности (ОКО). При этом данные показатели являются частными критериями многокритериального управления процессом КР, выбор значений которых связан с поиском Парето-оптимальных решений.

Цель работы – повышение эффективности управления процессом КР на основе принятия Парето-оптимальных решений.

Для производства бензина важно получение октанового числа *ON* не ниже минимально допустимого ON_0. Поэтому, в качестве первого критерия управления процессом КР принимаем критерий максимизации среднего нормированного по величине ON_0 значения октанового числа *ON* на интервале времени [0, *T*]:

$$I_1 = \frac{\overline{ON}(T)}{ON_0} = \frac{1}{T \cdot ON_0} \int_0^T ON(t)dt \to \max . \qquad (1)$$

Для производства бензина также важно производить затраты *Z* не выше максимально допустимых Z_0. Поэтому, в качестве второго критерия управления процессом КР используем критерий минимизации среднего нормированного по величине Z_0 значения затрат *Z* на интервале времени [0, *T*]:

$$I_2 = \frac{\overline{Z}(T)}{Z_0} = \frac{1}{T \cdot Z_0} \int_0^T Z(t)dt \to \min . \qquad (2)$$

Для объединения критериев управления процессом КР введем ОКО в виде линейной свертки частных критериев:

$$I = k_1 \cdot I_1 + k_2 \cdot I_2 = \frac{1}{T} \int_0^T \left[k_1 \cdot ON_0 \cdot \frac{1}{ON(t)} + k_2 \cdot \frac{1}{Z_0} \cdot Z(t) \right] dt \to \min , \qquad (3)$$

где k_1, k_2 – весовые коэффициенты, регулирующие относительную важность критериев I_1, I_2 в отношении принятия решений, $k_1 + k_2 = 1$.

Использование нормированных значений критериев связано с возможностью их совмещения на одной критериальной плоскости и поиска Парето-оптимальных решений (рис. 1). Значения частных критериев и ОКО определяются с использованием гибридной модели процесса КР [2].

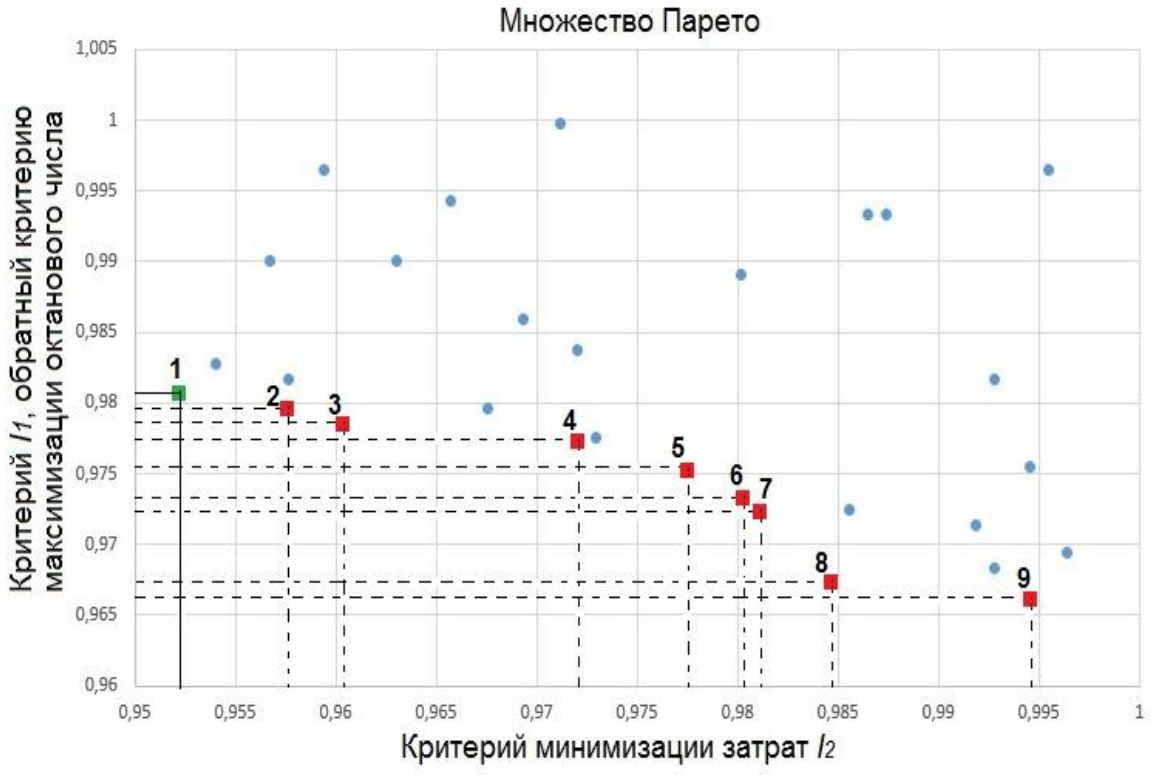


Рис. 1. Поиск оптимального решения на множестве Парето.

Множество Парето (рис. 1) получено на основе гибридной модели процесса КР [2] для переменных процесса КР [2], соответствующих допустимым интервалам изменения за ноябрь 2015 г. для установки ЛЧ-35-11/1000 при стандартных условиях протекания процесса КР [3].

Исходя из требования минимизации ОКО получено девять неулучшаемых решений в двумерном критериальном пространстве. Данные решения выбирались графически из условия, что точка, соответствующая решению, должна лежать ниже и левее всех остальных точек (рис. 1). Для проверки неулучшаемости данных решений из соответствующих точек (1–9) были проведены пунктиром лучи параллельно отрицательному направлению осей и получено, что в образованных углах других решений нет [4].

Получено множество неулучшаемых решений – множество Парето для принятия решений при управлении процессом КР:

$$\left\{I_1^1, I_2^1\right\}, \left\{I_1^2, I_2^2\right\}, ..., \left\{I_1^i, I_2^i\right\}, ..., \left\{I_1^9, I_2^9\right\}, \tag{4}$$

где $\left\{I_1^i, I_2^i\right\}, i = \overline{1,9}$ – значения частных критериев для *i*-ого неулучшаемого решения.

Из множества Парето (4) определено наилучшее решение, соответствующее минимуму ОКО:

$$\left\{I_1^1, I_2^1\right\} : I \to \min, \tag{5}$$

Оптимальное решение (5) соответствует значениям критериев: $I_1 = 0{,}981$; $I_2 = 0{,}952$; $I = 0{,}966$ при $k_1 = k_2 = 0{,}5$. На основе гибридной модели [5] для оптимального решения (5) получены: затраты на производство бензина $Z = 0{,}825$ (млрд руб.); октановое число продукта $ON = 92{,}83$.

Исходя из выражений (1–2) оптимальным при управлении процессом КР является поддержание среднего значения октанового числа бензина *ON* в течение одного месяца

на уровне 92,83, а среднего значения затрат на производство бензина – на уровне 0,825 (млрд руб.).

В результате сравнения с существующими показателями качества процесса КР процедура принятия Парето-оптимальных решений позволила достичь положительных эффектов: снижение затрат на 0,16 млн руб.; повышение октанового числа на 0,5; снижение значения ОКО на 0,044.

Библиографический список

1. Джамбеков, А. М. Нечеткий регулятор стабилизации катализата / А. М. Джамбеков, И. А. Щербатов // Современные технологии. Системный анализ. Моделирование. – 2016. – № 4 (52). – С. 106–114.

2. Джамбеков, А. М. Расчет эффективности процесса каталитического риформинга с использованием гибридной модели / А. М. Джамбеков, И. А. Щербатов, О. В. Антонов, О. М. Проталинский // Автоматизация, телемеханизация и связь в нефтяной промышленности. – 2017. – № 2. – С. 26–33.

3. Джамбеков, А. М. Локальный ПИД-регулятор стабилизации катализата / А. М. Джамбеков, И. А. Щербатов // Современные технологии. Системный анализ. Моделирование. – 2016. – № 3 (51). – С. 98–104.

4. Системный анализ в информационных технологиях : учеб. пособие / Ю. Ю. Громов, Н. А. Земской, А. В. Лагутин, О. Г. Иванова, В. М. Тютюнник. – 2-е изд., стереотип. – Тамбов : Изд-во Тамб. гос. техн. ун-та, 2007. – 176 с.

5. Джамбеков, А. М. Математическое описание качества процесса каталитического риформинга на основе гибридной модели / А. М. Джамбеков, И. А. Щербатов // Моделирование. Теория, методы и средства : материалы 16-й Междунар. науч.-практ. конф., посвящ. 110-летию Южно-Российского государственного политехнического университета (НПИ) им. М. И. Платова. – Новочеркасск : ООО «Лик», 2016. – С. 108–111.

Д. Н. Люлякина, О. Ю. Кузнецова

Пензенский государственный университет, Пенза, Россия

ОПТИМИЗАЦИЯ СИСТЕМЫ МЕНЕЖДМЕНТА ПРЕДПРИЯТИЯ: ПРОЦЕССНЫЙ ПОДХОД

На сегодняшний день конкурентоспособность многих фирм на рынке низка вследствие неэффективного использования принципов процессного подхода. Актуальность темы заключается в выявлении возможностей повышения прибыльности компании на основе оптимизации бизнес-процессов.

В настоящее время современная рыночная среда сложна, динамична и требует от компании систематического совершенствования своих систем управления, а также модернизации информационных систем их поддержки. На сегодняшний момент, общепризнанным фактом, является, что кризис экономики – это, прежде всего, кризис управления, из-за несоответствия качества менеджмента компаний требованиям сферы рынка. Выходом из такого положения является внедрение эффективных механизмов управления в соответствии с общепринятыми мировыми стандартами.

Наука управления в настоящее время по-прежнему занимается поиском эффективных форм использования организационных ресурсов на фоне усложняющейся экономической ситуации, путей выхода или недопущения критических ситуаций на основе предвиде-

ния. В этих условиях изменения требуется от процессов управления, особой мобильности и динамичности. Понимание того, что, не придавая значению роли процесса управления, это может привести к непроработанности, а во многих случаях и полному отсутствию действенных инструментов его эффективной организации, заставляет компании иначе взглянуть на процессный подход.

Во всем мире применение процессного подхода к управлению компанией считается одним из важнейших инструментов повышения эффективности бизнеса. Процессный подход – главная концепция менеджмента организации, которая сформировалась в 80-х г. прошлого века. Данная концепция подразумевает, что вся деятельность организации рассматривается как набор процессов. Для того чтобы управлять, необходимо управлять бизнес-процессами. Процессный подход является ключевым элементом улучшения качества ведения бизнеса.

Главное понятие, которое использует процессный подход – это понятие бизнес-процесса. Существует множество определений, но наиболее часто используется определение стандарта ISO 9001. «Процесс – это совокупность взаимосвязанных и взаимодействующих видов деятельности, которые преобразуют входы в выходы». Важной составляющей процесса, которая не отражена в этом определении, является систематичность действий. Действия процесса должны быть повторяющимися, а не случайными.

Процессный подход применяется с целью построения горизонтальных связей в организациях. Структурные подразделения и сотрудники компании, участвуют в бизнес-процессах, они способны самостоятельно решать возникающие проблемы без участия непосредственных руководителей и координировать работу в рамках процесса .

Благодаря данному подходу, можно более оперативно решать возникающие проблемы и воздействовать на конечный результат. Отличительной особенностью процессного подхода от функционального, позволяет концентрироваться не на работе каждого из структур подразделений, а на результатах работы компании в целом. В отличие от функционального подхода, управление процессами позволяет концентрироваться не на работе каждого из подразделений, а на результатах работы организации в целом. При процессном подходе, главным элементом становиться процесс, и понятие структуры организации уходит на второй план. Так как данный подход подразумевает, что организация состоит не из подразделений, а из бизнес-процессов.

Процессный подход опирается на множество принципов. Применение этих принципов в компании позволяет значительно повысить эффективность работы, увеличить скорость обработки данных и т.д. Однако, чтобы внедрить данные принципы, в компании должна быть сформирована высокая корпоративная культура. Процессный подход требует от работников организации постоянной совместной работы, несмотря на то, что они могут относиться к различным структурным подразделениям. От того, насколько удастся наладить совместную работу в коллективе, будет зависеть «работоспособность» принципов, лежащие в процессном подходе.

Наиболее популярными принципами процессного подхода являются:

- принцип востребованности процесса. У любого бизнес-процесса должна быть, а его результаты должны иметь внешних и внутренних потребителей;
- принцип взаимосвязи процессов. Компания, представляет собой сеть взаимосвязанных между собой бизнес-процессов;
- принцип контроля процесса. Бизнес-процесс имеет начало и конец, которые определяют границы показателей процесса, и его результат.

Принцип ответственности за процесс. В реализации бизнес-процесса могут быть задействованы различные сотрудники компании, но контролировать и нести ответственность за результат должен один человек. Владелец процесса – процессный подход использует это понятие как одно из самых важных. Владельцем является человек, имеющий в своем арсенале необходимое количество человеческих, материальных ресурсов и несущий ответственность конечный результат процесса.

- Принцип документирования процессов. Бизнес-процессы необходимо стандартизировать, документировать, т.е. получать все данные об изменениях и совершенствованиях деятельности процесса.

Процессный подход формирует горизонтальные связи в работе компании, благодаря существует ряд преимуществ, в отличии от функционального подхода.

Основными преимуществами процессного подхода являются:

- координация действий различных отделов в рамках бизнес-процесса;
- ориентация на результат процесса;
- повышение эффективности работы и конкурентоспособности организации;
- повышение предсказуемости результатов;
- выявление минусов в управлении деятельности организации с целью модернизации бизнес-процессов;
- минимизация издержек компании;
- исключение нецелесообразных бизнес-процессов;
- сокращение времени на обработку данных и материальных затрат.

Процессный подход лежит в основе часто используемых и достаточно эффективных методов по совершенствованию деятельности организации. На сегодняшний день выделяют четыре направления, которые используют процессный подход в качестве главного подхода по повышению эффективности работы организации.

К таким направлениям относятся:

1. Всеобщий менеджмент качества (TQM). Данная концепция, предусматривает непрерывное повышение качества выпускаемой продукции, бизнес- процессов и системы менеджмента организацией. В основу работы организации ставится удовлетворение потребностей потребителя.

2. Постоянное улучшение процессов (Continuous Improvement Process). Это концепция, которая предусматривает минимальные, но постоянные оптимизации процесса. Наиболее популярным подходом, в основе которого лежит постоянное улучшение процессов является японский подход «кайдзен» (kaizen);

3. Совершенствование бизнес-процессов (Business Process Improvement) или управление бизнес процессами (Business Process Management). Это подход, направленный на то, чтобы помочь организациям оптимизировать бизнес процессы с целью повышения их эффективности. Изменения процессов обязательность происходят на систематической основе.

4. Реинжиниринг бизнес-процессов (Business Process Reengineering). Данное направление основывается на переосмысление существующих процессов, кардинальных изменений их структуры (перепроектирование), а также на применение в большом количестве информационных технологий. В отличие от трех вышеописанных направлений реинжиниринг предусматривает быстрое изменение процессов.

Компания, построенная по процессному принципу, более гибкая на все воздействия со стороны внешней среды. Так как в настоящий момент в мире век информационных технологий, то любой компании необходимо понимать, что для того, чтобы находиться наплаву и быть конкурентоспособной организацией на рынке, необходимо постоянно улучшать и оптимизировать свои бизнес-процессы, а это возможно, если правильно внедрить процессный подход в компанию. Процессный подход значительно упрощает адаптацию новых сотрудников и снижает зависимость работы компании от человеческого фактора. Таким образом, многие проблемы современного российского менеджмента может решить использование процессного подхода и инструментов управления бизнес-процессами. Данная технология на сегодняшний момент является очень популярной, так как она позволяет навести порядок в компании и заложить механизм улучшения процессов.

Библиографический список

1. Менеджмент процессов : пер. с нем. / Й. Беккер, Л. Вилков, В. Таратухин, М. Кугелер, М. Роземанн. – М. : Эксмо, 2014.

2. Медынский, В. Г. Инновационный менеджмент : учеб. для вузов / В. Г. Медынский. – М. : ИНФРА-М, 2013.

3. Смирнов, Ю. Н. Основные бизнес-задачи предприятия / Ю. Н. Смирнов // Интеграл. – 2008. – № 3. – С. 96–98.

Д. А. Таланов

Пензенский государственный университет, Пенза, Россия

РАЗРАБОТКА АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ УПРАВЛЕНИЯ ЖИЛЫМ ПОМЕЩЕНИЕМ

Рассматривается автоматизация работы системы регулирования влажности воздуха в помещении. Разработка такой системы является очень актуальной задачей, поскольку установка аналогов таких систем стоит больших сумм по нескольким причинам – это и техническая сложность реализации системы и высокая стоимость комплектующих.

Системы «Умный дом» – это система устройств, предназначенных для решения задач без участия человека. Распространенные примеры – автоматическое регулирование работы отопительной системы, кондиционеров, уведомления о протечках воды, вторжении, возгорании в доме [1]. Автоматизация домашнего быта – удобный подход для конструирования и настройки необходимых параметров системы в зависимости от собственных потребностей.

В данной статье предполагается реализация автоматизированной системы управления жилым помещением. Автоматизированное управление осуществляется через средства контроля влажности в жилом помещении. Система предназначена для контроля и регулирования уровня влажности в доме таким образом, чтобы сохранять уютный микроклимат и сократить расходы на его поддержание.

В мире существует огромное решение систем типа «Умный дом», имеющие свои особенности, уровни надежности. Стоимость таких систем для разных производителей на один строительный объект различается в разы. И большинство из них имеют разные функциональные возможности и техническую реализацию. Для примера рассмотрим функцию «центрального выключателя». Уходя из дома, человек обесточивает все электроприборы, выключает все освещение, звонит на пульт охраны для сдачи помещения, если оно находится под управлением охранной системы. И только после этого закрывает и запирает дверь. Автоматизированный «центральный выключатель» сделает все выше перечисленное автоматически после нажатия на нужную клавишу. Таким образом, «центральный выключатель» служит командой для одновременного выполнения двух абсолютно разных функций – выключения электроприборов и включения охранной сигнализации.

Проведем анализ уже существующих подобных систем:

1) системы автоматизации компании «EasySmartBox»;
2) система управления «Умный дом» компании «Системы домашней автоматики»;
3) система «Умный дом» компании «Green Vision»;
4) система «Smart House» компании A .P. Technology.

Системы автоматизации компании «EasySmartBox» имеют недостатки – достаточно высокую стоимость (установка системы в помещении от 150 тыс. руб.) и отсутствие в заявленном функционале компонента для регулирования влажности в помещении.

Система управления «Умный дом» компании «Системы домашней автоматики» основана на использовании контроллеров «Ocelot» и «Leopard», которые на данный момент являются одними из самых надежных устройств управления домашней автоматикой в своей ценовой категории [2]. Данное техническое решение выглядит наиболее стоящим, однако высокая цена установки (от 250 тыс. руб.) снижает конкурентоспособность данной системы.

Система «Умный дом» компании «Green Vision» отличается легкостью установки, простотой эксплуатации, высокой надежностью. Это одно из самый технологичных реше-

ний, однако отсутствие среди заявленного функционала управления влажностью воздуха в помещении и очень высокая стоимость готовой продукции (от 5 тыс. евро за установку) также снижает конкурентоспособность системы.

Система «Smart House» компании A. P. Technology изучает проекты многофункциональных систем управления бытовой электроникой [3]. Система осуществляет управление практически любыми электронными устройствами. Однако очень высокая стоимость при установке системы (для квартиры до 100 кв.м. от около 750 тыс. руб.) снижает спрос покупателей.

Предлагаемое техническое решение состоит из аппаратной и программной части. Аппаратная часть – датчик управления средствами устранения влажности (например, вентилятор). Этот датчик получается из модуля взаимодействия с системой управления и непосредственным модулем управления средствами устранения влажности. Модуль взаимодействия – это Bluetooth-модуль с программным обеспечением, обеспечивающим логику управления. Программная часть системы разбивается на 2 компонента – программное обеспечение Bluetooth-модуля и серверная часть системы, которая позволяет хранить настройки датчика и выполнять общим управлением всей системы в целом. Программное обеспечение Bluetooth-модуля должно выполнять включение и выключение средств понижения влажности воздуха. Серверный компонент должен позволять осуществлять ввод верхней и нижней границ активации датчика контроля.

На рис. 1 приведена схема устройства управления средствами понижения влажности воздуха в помещении.

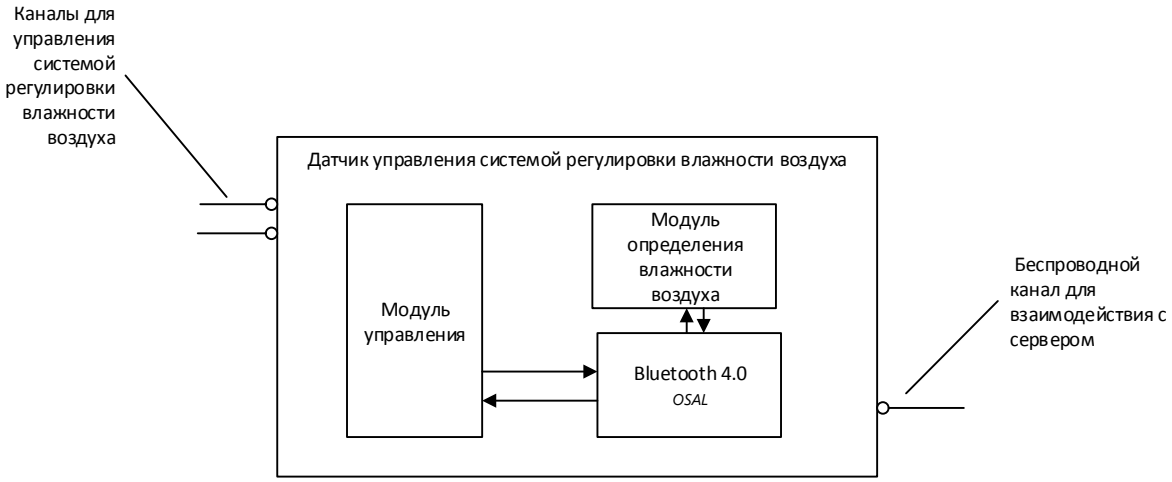


Рис. 1. Общая схема датчика управления системой регулировки влажности воздуха

Проектирование системы имеет особенности, в том числе и наличие у устройстве управления беспроводного датчика Bluetooth 4.0. Регулирование модуля Bluetooth осуществляет программный слой OSAL, осуществляющий взаимодействие с сервером, с модулем управления средствами понижения влажности воздуха в помещении и с модулем определения влажности воздуха в помещении. Bluetooth-модуль состоит из чипа CC 2540. Это решение от компании Texas Instruments, позиционируется как System-on-chip. Модуль Bluetooth версии 4.0 используется из-за появление в спецификации энергосберегающего режима, который актуален в устройствах предлагаемого типа. На рис. 2 приведен выбор границы отключения датчика.

В ходе реализации автоматизированной системы управления жилым помещением был выполнен анализ существующих аналогов, и были созданы функциональные требования к разрабатываемой системе, осуществлено проектирование программного обеспечения.

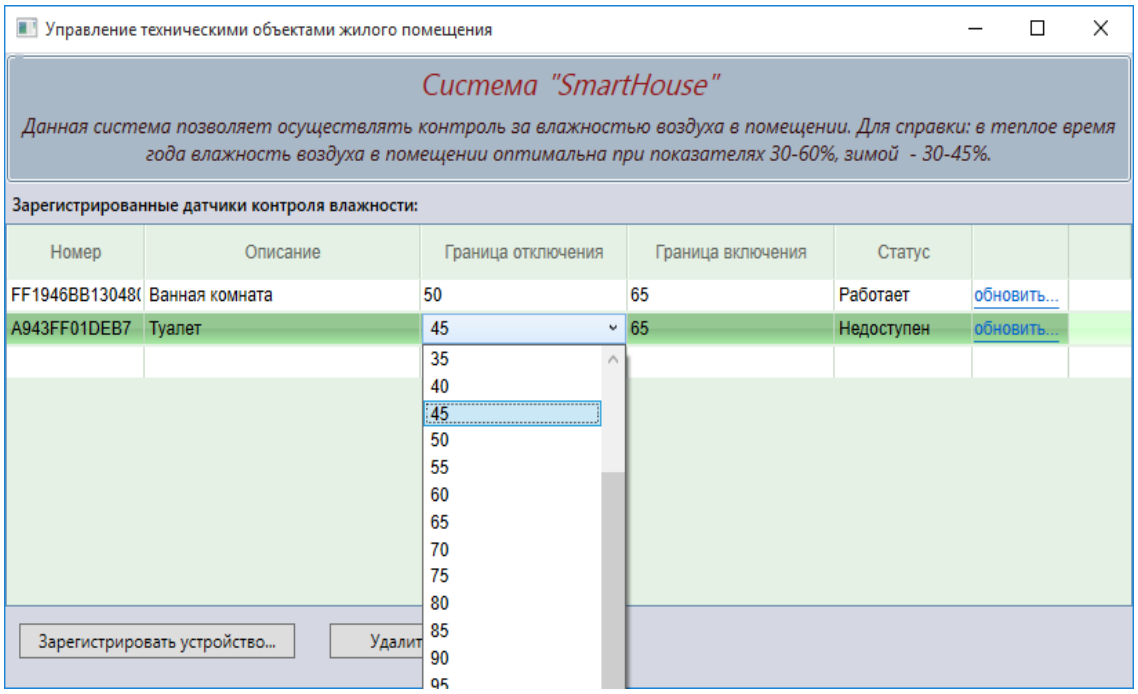


Рис. 2. Выбор границы отключения датчика

Проектная часть осуществляется с использованием объектно-ориентированного языка моделирования UML. Разработка программного обеспечения выполняется на языке программирования C# с применением платформы .Net Framework 4.5.1. Для создания системы использовалась среда разработки Microsoft Visual Studio 2015. Таким образом, разработка автоматизированной системы управления жилым помещением реализована в полном объеме.

Библиографический список

1. Умный дом. – URL: https://ru.wikipedia.org/wiki
2. Автоматизированная система управления «Умный дом» на базе контроллеров «Ocelot», «Leopard». – URL: http://smartchel.ru/old/article/3.php
3. Technology? A. P. – URL: http://www.aptech.ru/

А. Ю. Тимонин, А. С. Бождай

Пензенский государственный университет, Пенза, Россия

УРОВНИ МАТЕМАТИЧЕСКОГО ОПИСАНИЯ ПРОЦЕССА ПОСТРОЕНИЯ СОЦИАЛЬНОГО ПРОФИЛЯ

Рассматриваются аспекты моделирования социального профиля человека. Приведено описание абстрактно-алгебраического уровня модели социального профиля на основе теории графов, а также затрагиваются вопросы реализации логического и физического уровней модели представления данных социального профиля.

Социальный профиль (СП) – это множество, состоящее из информации, способной тем или иным образом охарактеризовать человека, причем эта информация должна быть

сгруппирована для удобства человеческого восприятия [4]. Задача социального профилирования относится к теоретической концепции анализа социальных сетей и использует схожие механизмы. Однако, в поставленной задаче помимо прямых связей между личностями на конечный результат влияние оказывают также и индивидуальные характеристики человека: его деятельность, увлечения, контактная информация, черты характера, общественный статус и так далее. Как и в анализе социальных сетей при создании СП для учета всех связей между его элементами разрабатывается математическая модель, а затем строится социальный граф [2], в котором узлы представлены социальными объектами (например, сообществами, медиа-контентом и пользовательскими профилями с различными атрибутами), а ребра – социальными связями между ними. На основе математической модели создается инфологическая модель базы данных социального профиля.

Для упрощения создания и эксплуатации сложных систем принято пользоваться математическими описаниями разных уровней. Для получения наиболее полных сведений важно изучить систему на всех целесообразных уровнях абстракции. В рамках текущей работы следует разделять и рассматривать отдельно модель системы построения социального профиля и модель, собственно, самого СП.

Лингвистический уровень описания, как наиболее высокий, используется для формализации объектов с целью последующей манипуляции с их моделями. В частности, на этом уровне можно описать процесс построения социального профиля, который задается рядом шагов:

1. Сбор исходной информации из разных источников для большей полноты итогового СП;
2. Фильтрация данных от бессмысленной, противоречивой и конфиденциальной информации;
3. Выделение из множества собранных данных уникальных ключевых сущностей, которые станут центральными узлами сети, и создание на их базе структурированной информационной карты [1];
4. Распределение динамического контента по его типу в соответствующие хранилища информации и дальнейший поиск социальных объектов как в текстовых, так и мультимедиа данных;
5. Выделение связей между узлами (как однонаправленных, так и двусторонних);
6. Расширение сети до состояния, приемлемого для последующего анализа, за счет еще не включенных связанных объектов;
7. Поиск существующих множественных связей;
8. Поиск аналогий в нерассмотренных динамических данных и включение их в описание созданных объектов сети для исключения противоречивости данных;
9. Постройка социального графа на основе полученных сведений.

Но более низком абстрактно-алгебраическом уровне задается описание математической модели СП. При ее создании, чья возможная структура показана на рис. 1, используется аппарат теории графов. Ввиду применения данной модели в нереляционных БД, проблема ее расширяемости отсутствует [4].

Представленные на рис. 1 группы объектов были выбраны вследствие их наглядности, доступности и большого числа вхождений в исходных данных в рамках текущей работы. Данный список условен и может быть расширен в случае необходимости создания социального профиля для конкретной задачи. Каждый из объектов данной модели представляет собой множество сущностей СП, которое состоит из выражений, подходящих под его описание, причем у каждой такой сущности есть уникальный набор атрибутов. Это позволяет проводить более гибкий анализ входных данных и обеспечивает эффективное использование результатов построения.

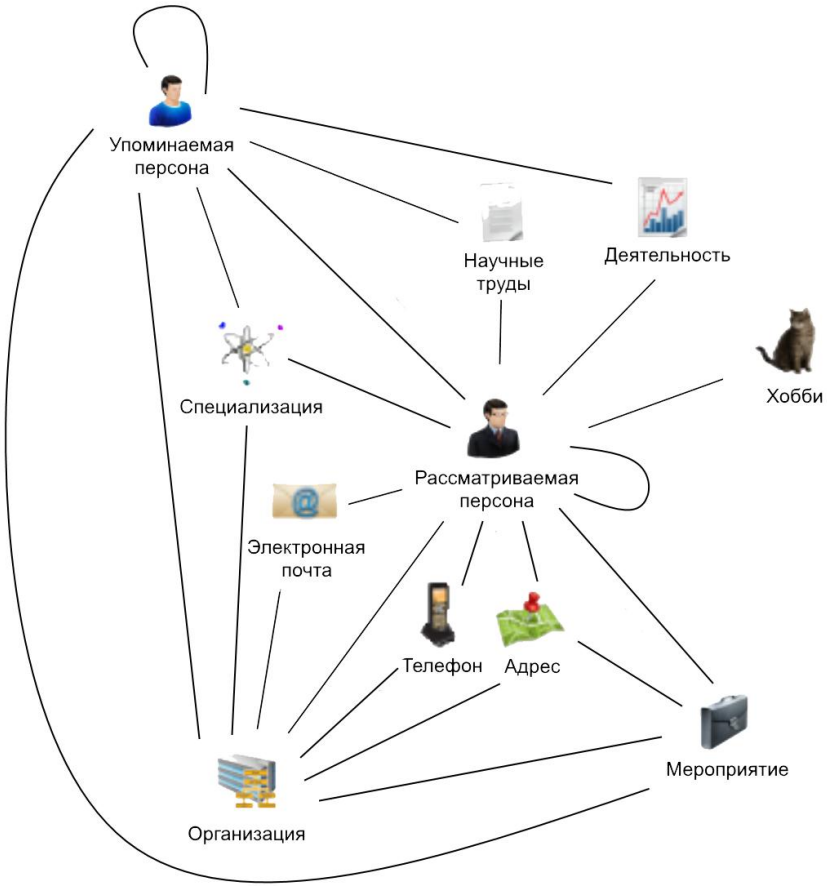


Рис. 1. Общая структура социального графа

Математическое представление социального профиля описывается следующей функцией [3]:

$$P(S,R) = \sum_{i=1}^{m} v_i S(X)_i + \sum_{j=1}^{n} u_j R(Y)_j \quad (1)$$

где $S = \{S_1, S_2, ..., S_m\}$ – множество социальных объектов;

$R = \{R_1, R_2, ..., R_n\}$ – множество связей между объектами;

X_i, Y_j – матрицы свойств для каждого объекта/связи СП;

m, n – количество объектов и связей соответственно, причем $m–1 \le n$;

v, $u \in [0;1]$ – весовые коэффициенты объектов и связей соответственно, определяющие значимость и достоверность приводимой информации.

Внутри исходных текстовых данных множество социальных объектов представлено существительными, числительными, словосочетаниями и символьными конструкциями, а множество связей – глаголами, причастиями, деепричастиями и пунктуационными знаками (дефис, двоеточие и т.д.). Значения весовых коэффициентов зависят от ряда факторов: авторитетность источника информации, количество упоминаний конкретной информации с привязкой ко времени, наличие доказательств в пользу истинности данных, отсутствие противоречивых и опровергающих сведений.

Указанная выше математическая модель служит основой для информационного обеспечения системы построения СП. Оно представлено реляционной базой данных для хранения информационной карты и связкой из распределенного хранилища HBase, графовой и NoSQL БД для динамического контента. Особых требований к хранению неструктурированной информации нет, и количество столбцов у собранных таким образом данных может не совпадать. Результаты анализа данных помещаются в iBase базу данных [5], на основе которой строится графовое представление социального профиля.

Фактически социальный граф является двудольным, но на практике множество связей изображается не в виде вершин, а подписанных линий. Они не играют серьезной роли в плане визуализации результатов анализа, но помогают разграничить воздействие одних объектов на другие. Связи, как и социальные объекты, могут иметь собственные наборы атрибутов. Возможные отношения между объектами и типами связей СП представлены в табл. 1.

Таблица 1

Пример матрицы инцидентности БД социального профиля

	Адресация	Связь с организацией	Специализация организации	Персоны на мероприятии	Связь с персоной	Хобби персоны	Деятельность персон	Персоны в организации	Контакты персон	Специализации персон	Привязка научных трудов
Рассматриваемая персона	1	0	0	–1	–1	1	1	1	±1	1	1
Адрес	–1	0	0	0	0	0	0	0	0	0	0
Деятельность	0	0	0	0	0	0	–1	0	0	0	0
Мероприятие	1	0	0	1	0	0	0	0	0	0	0
Научные труды	0	0	0	0	0	0	0	0	0	0	–1
Организация	1	–1	1	0	0	0	0	–1	0	0	1
Специализация	0	0	–1	0	0	0	0	0	0	–1	0
Телефон	0	1	0	0	1	0	0	0	0	0	0
Хобби	0	0	0	0	0	–1	0	0	0	0	0
Электронная почта	0	1	0	0	1	0	0	0	0	0	0
Упоминаемая персона	1	0	0	–1	0	0	1	1	±1	1	1

Разные знаки в ячейках матрицы указывают на две группы объектов, объединяющихся соответствующей связью. Направление связи в рамках заполнения социального профиля не имеет значения ввиду возможного наличия в исходных текстах различных языковых конструкций «подлежащее – сказуемое – дополнение». Знак «±» означает допустимость связей-петлей для однородных групп объектов.

При проектировании инфологической модели СП множества социальных объектов и связей разделяются на группы-подмножества и образуют структуры, включающие в себя массивы уникальных свойств, задающиеся списком атрибутов (табл. 2).

Таблица 2

Примеры атрибутов сущностей БД социального профиля

Сущность	Атрибут	Тип	Примечание
Деятельность	id_job	Идентификатор	
	job_name	Текст, обяз.	название
	job_begin	Текст	начало работы
	job_end	Текст	конец работы
Мероприятие	id_conf	Идентификатор	
	conf_name	Текст	название
	conf_date	Дата	время проведения
Рассматриваемая персона	id_person	Идентификатор	
	FIO	Текст, обяз.	имена, псевдонимы
	Age	Текст	возраст/дата рождения
Электронная почта	id_mail	Идентификатор	
	email	Текст, обяз.	адрес электронной почты
	domain	Текст	домен, на котором расположена почта
Рассматриваемая персона <–> Организация	post_org	Текст	занимаемая должность
	work_begin	Текст	начало работы человека в организации
	work_end	Текст	конец работы человека в организации

Также в табл. 2 приведены типы данных для физической реализации базы данных СП в IBM iBase. Ввиду неструктурированности вся проанализированная информация помещается в БД в виде текста, однако теоретически возможна ее конвертация к предпочитаемым типам данных (время, число, телефонный номер и т.д.).

Заключение

В рамках данной работы были рассмотрены лингвистический уровень описания процесса построения СП, абстрактно-алгебраический, инфологический и физический уровни описания социального профиля. Результатами являются лингвистическое описание процесса социального профилирования, математическая модель СП, пример инфологической модели графовой базы данных социального профиля и форматы данных для хранения анализируемой информации. Выходными данными системы являются практические результаты профилирования в виде социального графа.

Библиографический список

1. Timonin, A. Y. The Process of Personal Identification and Data Gathering Based on Big Data Technologies for Social Profiles / A. Y. Timonin, A. M. Bershadsky, A. S. Bozhday // First International Conference «Digital Transformation and Global Society (DTGS)». – St. Petersburg, Russia, June 22–24, 2016. – Vol. 674. – P. 576–584.

2. Анализ социальных сетей // Материал из Википедии – свободной энциклопедии. – URL: https://ru.wikipedia.org/wiki/Анализ_социальных_сетей (дата обращения: 24.08.2017).

3. Тимонин, А. Ю. Визуализация данных социального профиля для использования в прикладных задачах / А. Ю. Тимонин // Вестник Пензенского государственного университета. – 2017. – № 1 (17).

4. Тимонин, А. Ю. Разработка социального профиля на основе имеющихся инструментальных средств / А. Ю. Тимонин // Новые информационные технологии и системы : сб. науч. ст. XII Междунар. науч.-техн. конф. – Пенза : Изд-во ПГУ, 2015. – С. 221–224.

5. Шмид А. В. Новые методы работы с большими данными: победные стратегии управления в бизнес-аналитике / А. В. Шмид. – М. : ПАЛЬМИР, 2016. – 528 с.

Р. А. Тураев

Пензенский государственный университет, Пенза, Россия

ПРИМЕНЕНИЕ WEB-ТЕХНОЛОГИЙ В РАЗРАБОТКЕ ИНФОРМАЦИОННЫХ СИСТЕМ

Рассматривается концепция разработки информационных систем с применением web-технологий, обозреваются существующие языки программирования, анализируются их средства разработки систем.

В начале XXI в. для человеческой цивилизации стартовала новая эпоха развития – эпоха информационного общества, которая характеризуется развертыванием новейшей информационно-телекоммуникационной революции, быстрым распространением информационных технологий, глобализацией общественных процессов и многопрофильной кооперацией. Происходит формирование новой глобальной информационно-коммуникационной среды жизни, образования, общения и производства, которая называется инфосфера. Организационно-технологическую основу информационного общества составляют глобальные информационные сети, ядром которых является Интернет.

Телекоммуникационная революция создает новые благоприятные возможности для формирования и развития сетевых структур в различных областях общественной жизни, включая экономику, безопасность, образование, медицину, культуру, науку, политику.

Одним из средств разработки информационных систем являются Web-технологии, которые набирают популярность в настоящее время [1].

Для работы Web нужна сеть TCP/IP. Это заставляет отказываться от других сетевых технологий не только в глобальных сетях, но и в сетях локальных. С другой стороны, наличие TCP/IP делает проще интеграцию информационной системы с другими информационными сервисами, например, электронной почтой.

При этом решения автоматически становятся масштабируемыми, так как для технологий TCP/IP нет никакой разницы в том, где находятся ресурсы и интерфейс пользователя: локально, в рамках корпоративной сети, или распределены по глобальной сети.

Двигаясь в этом направлении, очевидным является тот факт, что основной областью применения Web-технологий становятся распределенные информационные системы.

При этом должны унифицироваться форматы подготовки, обмена и отображения информации.

Рассмотрим процесс миграции информационной системы из традиционной технологической схемы локальной системы в Web-технологию.

Основными элементами традиционной схемы являются:

- интерфейс пользователя;
- ядро системы;
- информационный массив;
- интерфейс администратора;
- утилиты администратора.

С точки зрения Web-технологии интерфейс пользователя – это браузер, который взаимодействует с ядром через http-сервер. Таким образом происходит первый этап декомпозиции традиционной информационной системы в Web.

Второй шаг – это возможность использования браузера в качестве интерфейса администратора. Здесь появляются вопросы разграничения доступа и актуализации информации в базах данных системы.

Следующий шаг – распределение нагрузки по нескольким серверам, а также использование кэширования на серверах-посредниках [2].

Пока декомпозиции подвергалась связка «конечный пользователь – ядро». Теперь необходимо провести декомпозицию и на стороне сервера. Первым таким шагом является применение CGI (Common Gateway Interface – «общий интерфейс шлюза») при доступе к ресурсам. Сервер становится посредником между браузером и сервером ресурса.

Более эффективно это можно решить за счет API (Application Programming Interface – «программный интерфейс приложения»), когда сам http-сервер имеет модуль доступа к backend процессу.

Не менее важным моментом является внедрение результатов обращения к внешнему ресурсу в готовые шаблоны страниц. В терминологии Web – это Server Site Include. Вставка происходит как локально, так и с использованием данных удаленного сервера.

Таким образом, сервер оснащается языком манипулирования данными при формировании отклика.

Другой инструмент – это видоизменение страниц на стороне клиента. Управление формой интерфейса пользователя осуществляется на стороне клиента при помощи скриптовых языков. Это, в первую очередь, влияет на повышение интерактивности информационной системы, построенной на основе Web-технологии.

Для того чтобы поддерживать реальный сеанс работы с сервером, в Web применяется спецификация Cookie. Идея заключается в том, чтобы передавать от клиента на сервер и обратно информацию о пользователе и его действиях, которая привязывается по типу информационного ресурса и времени [3].

На основании ряда аналитических статей в области Web-технологий за 2016–2017 гг., для более детального рассмотрения было выбрано несколько языков программирования, которые используются при разработке информационных систем [4].

PHP – это распространенный язык программирования общего назначения с открытым исходным кодом. PHP сконструирован специально для ведения Web-разработок и его код может внедряться непосредственно в HTML [5].

Ruby – динамический язык программирования с открытым исходным кодом с упором на простоту и продуктивность. Он обладает элегантным синтаксисом, который приятно читать и легко писать [6].

Python – высокоуровневый язык программирования общего назначения, ориентированный на повышение производительности разработчика и читаемости кода [7].

В результате сравнительного анализа возможностей языков программирования выявлены их преимущества и недостатки при разработке информационных систем (табл. 1, 2).

Таблица 1

Преимущества каждого из языков

PHP	Ruby	Python
Бесплатное программное обеспечение по лицензии PHP	Кроссплатформенность и открытый исходный код	Легкий и быстрый в изучении
Легкий в освоении (высокая скорость обучения)	Может быть встроен в язык разметки гипертекста	Поддерживается множеством платформ и операционных систем
Крупное сообщество пользователей и разработчиков	Язык программирования сверхвысокого уровня (VHLL)	Читабельный и организованный синтаксис
Расширенная поддержка баз данных	Простой и понятный синтаксис, позволяющий начинающему разработчику очень быстро изучить язык	Обеспечение быстрого прототипирования и динамических семантических свойств
Предоставляет большое количество доступных расширений и исходных кодов	Простое подключение к базам данных DB2, MySQL, Oracle и Sybase	Огромное сообщество поддержки
Разрешает исполнение кода в ограниченных средах выполнения	Созданные на Ruby большие масштабируемые программы просты в сопровождении	Простое построение приложений путем тестирования и импорта необходимых функций
Предоставляется возможность управления нативными сессиями и расширения API	Наличие встроенного отладчика и гибкого синтаксиса	Реюзабилити (возможность повторного использования) за счет тщательной разработки пакетов и модулей
Хорошая альтернатива конкурентам, таким как ASP (Active Server Pages) от Microsoft	Возможность написания внешних библиотек на Ruby или C	Объектно-ориентированный подход к программированию
Работает практически на любой операционной системе или платформе	Возможность написания многопоточных приложений с простым API	

Таблица 2

Недостатки каждого из языков

PHP	**Ruby**	**Python**
Не подходит для разработки настольных приложений	Возможны трудности в изучении	Недостаточно эффективная работа с многоядерными и многопроцессорными вычислительными системами
Традиционно скудный функционал для обработки ошибок	Нехватка информационных ресурсов	Ограниченный уровень доступа к базам данных
Глобальные параметры конфигурации могут менять семантику языка, осложняя процессы внедрения и совместимости	Большие затраты процессорного времени (CPU time) по сравнению с другими языками	Отсутствие коммерческой поддержки даже для Open Source проектов (однако, эта ситуация начинает меняться)
Обращение к объектам по умолчанию осуществляется методом «вызова по значению» (CallByValue), что противоречит аналогичным операциям для большинства языков и застает многих программистов врасплох	Сравнительно медленная разработка обновлений	Небольшое количество разработчиков Python по сравнению с другими языками, например, с Java
В целом считается менее защищенным по сравнению с другими языками программирования		По отзывам более медленный по сравнению с языками типа Java

Таким образом можно сделать следующие выводы:

- сценарии на языке PHP могут выполняться под всеми основными операционными системами. Большинство современных веб-серверов поддерживают PHP: Apache, Personal Web Server, Microsoft Internet Information Server, iPlanet, Netscape, Caudium, Oreilly Website Pro, OmniHTTPd, Xitami и другие. Язык PHP позволяет организовать обработку данных форм, создание динамических страниц сайта, создание и обработку файлов cookies и многие другие функции CGI (Common Gateway Interface);
- Python имеет легко читаемый синтаксис, что позволяет легко читать чужой код, разбираться в давно написанном собственном коде, огромное количество библиотек с кодом на любой случай жизни: будь то работа с таблицами Excel, изображениями или сетью Twitter;
- Ruby используется в прототипировании, в администрировании, в научных исследованиях, а также в разработке новых игр и разного рода приложений в качестве языка, необходимого для написания скриптов и т.д. Ruby не является столь безупречным продуктом, поскольку самый основной недостаток – это огромный объем ресурсов, который он потребляет и скорость работы.

Библиографический список

1. Румянцева, Е. Л. Информационные технологии / Е. Л. Румянцева, В. В. Слюсарь. – М. : ИД-форум : ИНФРА-М, 2007. – 256 с.
2. Сенченко, П. В. Web-ориентированные информационные технологии / П. В. Сенченко, Е. С. Павинич // Доклады ТУСУРа. – 2010. – № 2 (18) . – Ч. 2.
3. Применение Web-технологий. – URL: http://citforum.ru/seminars/cis99/useweb.shtml (дата обращения: 15.09.2017).
4. TIOBE Index for September 2017. – URL: https://www.tiobe.com/tiobe-index/ (дата обращения: 14.09.2017).
5. Официальный сайт PHP. – URL: http://php.net/manual/ru/intro-whatis.php (дата обращения: 13.09.2017).
6. Официальный сайт Ruby. – URL: https://www.ruby-lang.org/ru/ (дата обращения: 14.09.2017).
7. Официальный сайт Python. – URL: https://www.python.org/ (дата обращения: 13.09.2017).

З. И. Баусова, А. Ю. Старикова, Э. Ф. Шадрина

Пензенский государственный университет, Пенза, Россия

АНАЛИЗ СТАТИСТИЧЕСКИХ ПОКАЗАТЕЛЕЙ ПАТЕНТНЫХ ДАННЫХ

Содержатся результаты расчета статических показателей данных об изобретениях по формулам абсолютного прироста, темпов роста и темпов прироста. В работе представлены результаты расчета на основе данных Федеральной службы государственной статистики.

Имеются следующие данные о выдаче охранных документов (патентов) по России за 2005–2015 г. (в сопоставимых количествах), шт. (табл. 1) [1]:

Необходимо определить аналитические показатели динамики временного ряда с данными о выдаче патентов на изобретения в России за 2005–2015 гг.:

1) абсолютные приросты;

2) темпы роста;

3) темпы прироста;

4) абсолютное значение одного процента прироста, а также средние обобщающие показатели ряда динамики.

Таблица 1

Данные о выдаче патентов на изобретения по годам

Годы	Данные по количеству изобретений
2005	23390
2006	23209
2007	23028
2008	28808
2009	34824
2010	30322
2011	29999
2012	32880
2013	31638
2014	33950
2015	34706

Решение задачи. В зависимости от задачи исследования абсолютные приросты (y), темпы роста (T_p) и темпы прироста ($T_{\text{пр}}$) могут быть исчислены с использованием переменной базы сравнения (цепные) и постоянной базы сравнения (базисные).

1. Абсолютный прирост (y) – это разность между последующим уровнем ряда и предыдущим (или базисным).

Так, в 2006 г. прирост патентов был равен: цепной – $y_i = y_i - y_{i\text{-}1} = 23209 - 23390 = -181$ шт.; в 2007 г. – $23028 - 23209 = -181$ шт.

Аналогично исчисляются абсолютные приросты за любой год. Так абсолютный прирост базисный в 2006 г. составил $y_i = y_i - y_0 = 23390 - 23390 = -181$ шт.; а в 2007 г. $23028 - 23390 = -362$ шт. и т.д.

Таблица 2

Аналитические показатели патентных данных

Год	Патенты в сопост. кол-вах, шт.	Абсолютные приросты, шт.		Темпы роста, %		Темпы прироста, %		Абсолют. знач. 1 % прироста, шт.
		Цепн. (ежегод.)	Базисн. (к 2005 г.)	Цепн. (ежегод.)	Базисн. (к 2005 г.)	Цепн. (ежегод.)	Базисн. (к 2005 г.)	
А	1	2	3	4	5	6	7	8
05	23390	–	–	–	100 %	–	–	–
06	23209	–181	–181	99 %	99 %	0	0	–
07	23028	–362	–362	99 %	98 %	0	0	–
08	28808	5418	599	125 %	101 %	23,5 %	2,6	230
09	34824	11434	11434	120 %	148 %	39,7 %	48,9	288
10	30322	6932	6932	87 %	129 %	19,9 %	29,6	348
11	29999	6609	6609	98 %	128 %	21,8 %	28,3	303
12	32880	9490	9490	109 %	102 %	31,6 %	40,6	300
13	31638	8248	8248	96 %	135 %	25 %	35,6	329
14	33950	10200	10741	107 %	145 %	32,2 %	45,9	316
15	34706	11316	11316	102 %	148 %	33,3 %	48,4	339

Средний абсолютный прирост исчисляется двумя способами:

а) как средняя арифметическая простая годовых (цепных) приростов

$$\Delta\acute{o} = \frac{\sum \Delta\acute{o}}{\ddot{i}} =$$

$$= \frac{-181 - 362 + 5418 + 11434 + 6932 + 6609 + 9490 + 8248 + 10200 + 11316}{10} = \frac{69104}{10}$$

$= 6910{,}4$ шт.

б) как отношение базисного прироста к числу периодов

$$\Delta\acute{o} = \frac{\acute{o}_г - \acute{o}_0}{ï} = \frac{34706 - 23390}{10} = 1131{,}6 \text{ шт.}$$

2. Темп роста (T_p) – отношение уровней ряда динамики, которое выражается в коэффициентах и про центах. Цепной темп роста исчисляют отношением последующего уровня к предыдущему: $T_ц = y_i/y_{i-1}$; базисный – отношением каждого последующего уровня к одному уровню, принятому за базу сравнения: $T_б = \frac{y_i}{y_0}$.

Цепные темпы роста составили:

В 2006 г. по сравнению с 2005 г. $\acute{o}_{\frac{06}{05}} = \frac{23209}{23390} = 0{,}99$ (99 %)

В 2007 г. по сравнению с 2006 г. $\acute{o}_{\frac{07}{06}} = \frac{23\,028}{23209} = 0{,}99$ (99 %)

В 2008 г. по сравнению с 2007 г. $\acute{o}_{\frac{08}{07}} = \frac{28\,808}{23028} = 1{,}25$ (125 %)

В 2009 г. по сравнению с 2008 г. $\acute{o}_{\frac{09}{08}} = \frac{34\,824}{28\,808} = 1{,}2$ (120 %)

В 2010 г. по сравнению с 2009 г. $\acute{o}_{\frac{10}{09}} = \frac{30322}{34\,824} = 0{,}87$ (87 %)

В 2011 г. по сравнению с 2010 г. $\acute{o}_{\frac{11}{10}} = \frac{29999}{30322} = 0{,}98$ (98 %)

В 2012 г. по сравнению с 2011 г. $\acute{o}_{\frac{12}{11}} = \frac{32880}{29999} = 1{,}09$ (109 %)

В 2013 г. по сравнению с 2012 г. $\acute{o}_{\frac{13}{12}} = \frac{31638}{32880} = 0{,}96$ (96 %)

В 2014 г. по сравнению с 2013 г. $\acute{o}_{\frac{14}{13}} = \frac{33950}{31638} = 1{,}07$ (107 %)

В 2015 г. по сравнению с 2014 г. $\acute{o}_{\frac{15}{14}} = \frac{34706}{33950} = 1{,}02$ (102 %)

Базисные темпы за эти же периоды равны:

$\acute{o}_{06/05}$ =23209/23 390 = 0,99 (99 %); $\acute{o}_{07/05} = \frac{23\,028}{23\,390} = 0{,}98$ (98 %)

$\acute{o}_{08/05}$ =28 808/23 390 = 1,01 (101 %); $\acute{o}_{09/05} = \frac{34\,824}{23\,390} = 1{,}48$ (148 %)

$\acute{o}_{10/05}$ =30 322/23 390 = 1,29 (129 %); $\acute{o}_{11/05} = \frac{29999}{23\,390} = 1{,}28$ (128 %)

$\acute{o}_{12/05}$ =23880/23 390 = 1,02 (102 %); $\acute{o}_{13/05} = \frac{31638}{23\,390} = 1{,}35$ (135 %)

$\acute{o}_{14/05}$ = 33950/23 390 = 1,45 (145 %); $\acute{o}_{15/05} = \frac{34706}{23\,390} = 1{,}48$ (148 %)

Между цепными и базисными темпами роста имеется взаимосвязь: произведение соответствующих цепных темпов роста равно базисному. Зная базисные темпы, можно исчислить цепные делением каждого последующего базисного темпа роста на каждый предыдущий.

3. Темп прироста ($T_{пр}$) определяют двумя способами:

а) как отношение абсолютного прироста к предыдущему уровню:

$T_{пр} = \frac{\Delta y_i}{y_{i-1}}$ – цепной, или к базисному уровню $T_{пр} = \frac{\Delta y_i}{y_0}$ – базисный.

Расчет для цепного будет выглядеть так:

$\acute{o}_{06/05} = \frac{-181}{23390} = -0{,}007 \approx 0$; $\acute{o}_{07/06} = \frac{-362}{23209} = -0{,}0155 \approx 0$;

$\acute{o}_{08/07} = \frac{5418}{23028} = 0{,}235278$ (23,5 %); $\acute{o}_{09/08} = \frac{11434}{28808} = 0{,}3969$ (39,7 %) и т.д.

Расчет для базисного будет выглядеть так:

$\acute{o}_{06/05} = \frac{-181}{23390} \approx 0$; $\acute{o}_{07/05} = \frac{-362}{23390} \approx 0$; $\acute{o}_{08/05} = \frac{5418}{23390} = 0{,}0256$ (2,6 %);

$\acute{o}_{09/05} = \frac{11434}{23390} = 0{,}48884$ (48,9 %) и т.д.

б) как разность между темпами роста и единицей, если темпы роста выражены в коэффициентах: $T_{пр} = T_p - 1$; или как разность между темпами роста и 100 %, если темпы роста выражены в процентах: $T_{пр} = T_p - 100$ %.

$$\text{Т}_{\text{пр}08/09} = 1{,}2 - 1 = 0{,}2;\ \text{Т}_{\text{пр}\frac{08}{09}} = 120\ \% - 100\ \% = 20\ \%\ \text{и т. д.}$$

4. Абсолютное значение одного процента прироста (А1%) равно отношению абсолютного прироста цепного к темпу прироста цепному.

2007 г. – $A1\ \% = \frac{\Delta y}{T_{\text{пр}}} = \frac{5418}{23{,}5} = 230$ шт.

2008 г. – $A1\ \% = \frac{\Delta \text{y}}{\text{Т}_{\text{пр}}} = \frac{11434}{39{,}7} = 288$ шт. и т.д.

Расчет среднего абсолютного значения одного процента прироста за несколько лет производится по формуле:

$$\text{А 1\%ср} = \frac{\sum \text{А\%1}}{n}$$

$$\text{А1\%ср} = \frac{0 + 0 + 0 + 230 + 288 + 348 + 303 + 300 + 329 + 316 + 339}{11} = \frac{2453}{11} = 223\ \text{шт.}$$

Примером эконометрического моделирования является интервальный временной ряд количества изобретений в Пензенской области за 10 лет (с 2005 по 2015 г.) [2, 3]. Для того чтобы выбрать основную тенденцию изменения временного ряда (тренд) была проведена подгонка кривых. Были исследованы следующие модели кривых роста: линейная, логарифмическая, кубическая, квадратичная, роста и экспоненциальная. Слайд 15 В результате был выбран линейный тренд y =110018,5 + 9699,1 × t, так как он имел наименьшее отклонение от реальных данных. График модели с границами интервала прогноза представлен на рис. 1.

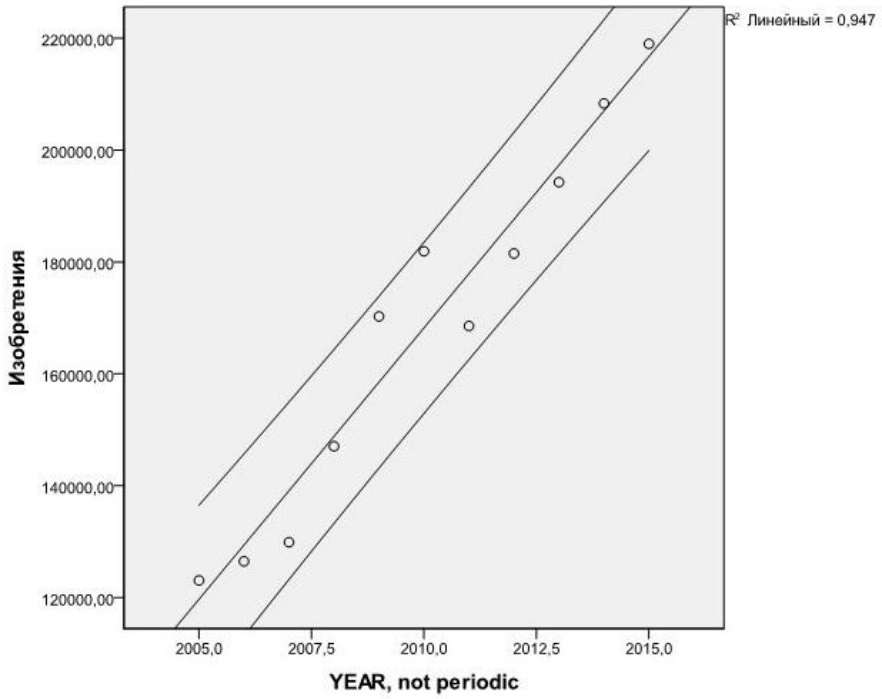


Рис. 1. График модели с границами интервала прогноза

Путем подстановки значений в уравнение кривой времени t, соответствующей периоду упреждения, был получен прогноз по изобретениям в России на 2016 и 2017 г.

Таким образом, по данным исследования можно сделать вывод, что ежегодно количество изобретений в России возрастает в среднем на 9699. Результаты прогноза за 2016 и 2017 г. соответственно равны: точечный прогноз 226 и 236 изобретений с нижними границами 209 и 218, верхними границами 244 и 254 изобретений с достоверностью 95 % и с вероятностью ошибки менее 0,001 [4–6].

Библиографический список

1. Баусова, З. И. Применение эконометрического моделирования к анализу патентных изобретении / З. И. Баусова, О. В. Прокофьев, А. Ю. Старикова, Э. Ф. Шадрина // Новые информационные технологии и системы : сб. науч. ст. XIII Междунар. науч.-техн. конф. – Пенза : Изд-во ПГУ, 2016. – С. 51–53.

2. Баусова, З. И. К вопросу о применении имитационного моделирования в выпускной квалификационной работе бакалавров (по направлению «прикладная информатика») / З. И. Баусова, Г. В. Бобрышева, А. Ю. Старикова, О. В. Прокофьев // Университетское образование (МКУО–2016) : сб. ст. XX Междунар. науч.-метод. конф. – Пенза : Изд-во ПГУ, 2016. – С. 84–85.

3. Шадрина, Э. Ф. К вопросу выбора прикладных программ для решения эконометрических задач в области интеллектуальной собственности / Э. Ф. Шадрина, В. О. Прокофьев // Информационные технологии в науке и образовании. Проблемы и перспективы : IV ежегодной межвуз. науч.-практ. конф. – Пенза : Изд-во ПГУ, 2017. – С. 226–228.

4. Старикова, А. Ю. К вопросу применения имитационного моделирования для управления интеллектуальной собственностью / А. Ю. Старикова, З. И. Баусова, Э. Ф. Шадрина // Новые информационные технологии и системы (НИТиС–2015) : сб. науч. ст. XII Междунар. науч.-техн. конф. – Пенза : Изд-во ПГУ, 2015. – С. 264–265.

5. Шадрина, Э. Ф. Оценка эффективности работы системы контроля и учета результатов интеллектуальной деятельности / Э. Ф. Шадрина, З. И. Баусова, А. Ю. Старикова // Информационные технологии в науке и образовании. Проблемы и перспективы : сб. науч. ст. III ежегодной межвуз. студ. науч.-практ. конф. – Пенза : Изд-во ПГУ, 2016. – С. 147–149.

6. Баусова, З. И. Оценка эффективности экономической информационной системы обработки агентских платежей за энергопотребление / З. И. Баусова, А. Ю. Старикова Э. Ф. Шадрина, Г. О. Звозникова // Информационные технологии в науке и образовании. Проблемы и перспективы : сб. науч. ст. IV ежегодной межвуз. науч.-практ. конф. – Пенза : Изд-во ПГУ, 2017. – С. 120–122.

И. Г. Кревский, А. В. Антонов

Пензенский государственный университет, Пенза, Россия

ЭЛЕКТРОННАЯ ИНФОРМАЦИОННО-ОБРАЗОВАТЕЛЬНАЯ СРЕДА ВУЗА КАК КЛЮЧЕВОЙ ЭЛЕМЕНТ ЕГО ИНФОРМАТИЗАЦИИ

Рассмотрены основные функции и возможности, реализованные в ЭИОС ПГУ. Рассмотрены вопросы интеграции ЭИОС с внешними информационными системами, а также создание модулей ЭИОС, расширяющих ее функционал на сферы деятельности университета, не входящие в собственно учебную деятельность.

В 2017 г. в Пензенском государственном университете введена в строй новая электронная информационно-образовательная среда вуза (ЭИОС). Она развернута на основе программного обеспечения с открытым кодом (LMS Moodle) и CMS собственной разработки (PHP+MySQL). Разработанные решения являются гибкими, масштабируемыми и адаптируемыми к изменению структуры вуза и программ его подготовки. Структура разработанной системы представлена в ряде публикаций, например, в [1]. При ее реализации выполнены все требования ФГОС ВО. Разработанная ЭИОС имеет следующие особенности.

1. Всем участникам учебного процесса предоставлен доступ к учебным планам, рабочим программам дисциплин, к изданиям электронных библиотечных систем (ЭБС) и электронным образовательным ресурсам (ЭОР), указанным в рабочих программах, а также другим материалам учебно-методических комплексов по каждой дисциплине (фонды оценочных средств, методические и учебные пособия и указания, компьютерные тесты и т.д.). Для каждой учебной дисциплины создан раздел в LMS Moodle, где размещены как непосредственно ресурсы по дисциплине, так и гиперссылки на материалы, размещенные в электронной библиотеке ПГУ, внешних ЭБС, с которыми университет имеет договора на использование их ресурсов, открытые образовательные ресурсы, а также карточки печатных изданий в электронном каталоге ИРБИС библиотеки вуза. Интерфейс заполнения курсов в ЭИОС достаточно простой, поэтому почти все размещение ресурсов выполняли непосредственно преподающие соответствующие дисциплины преподаватели. Была разработана стандартная структура курса, ориентированная на логичное и полное размещение учебно-методического комплекса по дисциплине, однако преподаватели могут менять эту структуру, предлагая авторскую.

2. В ЭИОС осуществляется фиксация хода образовательного процесса, включая учет текущей работы студентов в рамках балльно-рейтинговой системы, результатов промежуточной аттестации и результатов освоения основных образовательных программ (подсистема электронный деканат).

3. Реализована поддержка проведения всех видов занятий, процедур оценки результатов обучения, осуществляемых с применением электронного обучения (ЭО), дистанционных образовательных технологий (ДОТ).

4. Обеспечено формирование электронного портфолио обучающегося, в том числе, сохранение работ обучающегося, рецензий и оценок на эти работы. Сведения о направлении и профиле подготовки студента, результатах изучения им дисциплин автоматически поступают в портфолио из электронного деканата. Помимо представления результатов обучения и достижений студентов по ходу учебы, для поддержки трудоустройства выпускников, установлен нормативный срок хранения портфолио в открытом доступе на портале вуза в течение трех лет после завершения обучения.

5. Реализовано взаимодействие между участниками образовательного процесса. Синхронное взаимодействие в рамках обучения базируется на сервисе мгновенных сообщений. Также возможно подключение внешних модулей, например, для организации видеоконференций. Для асинхронного взаимодействия используются образовательные форумы в рамках каждой дисциплины (по выбору преподавателя форум сможет быть один или несколько, например, посвященные разным тематикам). Подсистема мониторинга ЭИОС обеспечивает организацию и проведение опросов среди студентов и преподавателей вуза, а также обработку их результатов. Благодаря интеграции с электронным деканатом, соответствующие опросы студенты получают с привязкой к изученным дисциплинам и реализующим их преподавателям.

6. Реализован инструментарий работников деканатов для работы с контингентом студентов (формирование групп, ведение личных и учебных карточек, поиск информации о студентах, формирование различных списков и т.д.).

При реализации ЭИОС были решены проблемы работы с высокой нагрузкой (зарегистрированы более 23 000 студентов и 3000 преподавателей и сотрудников, система устойчиво функционирует при одновременной работе более 7000 пользователей). В настоящее время в ЭИОС развернуто более 500 учебных планов (с учетом основных и переходных планов по многим образовательным программам), создано и заполнено около 24 000 учебных курсов. Разработаны решения, обеспечивающие массовую загрузку и, при необходимости, редактирование учебных планов в ЭИОС. Для мониторинга хода заполнения ЭИОС был разработан специализированный программный компонент, с помощью которого любой сотрудник вуза может в удобном формате в режиме онлайн получить информацию о заполнении курсов в ЭИОС. Также на основе этого компонента еженедельно формировались и рассылались руководителям от уровня заведующих кафедрами и выше рейтинги загрузки подразделениями УМК по закрепленным за ними дисциплинам в ЭИОС. Далее эти рейтинги оперативно рассматривались на ректорате и заседаниях советов факультетов, институтов и университета, что позволило обеспечить достижение 90 % загрузки ЭИОС за 8 месяцев.

Так как образовательная деятельность является основной функцией вуза, вокруг которой строится вся его работа, ЭИОС естественным образом выступает центральным элементом информатизации вуза. Осуществляется интеграция с ранее существовавшими в ПГУ информационными системами (ИС). ИС «Абитуриент» модернизирована и теперь использует единую с ЭИОС Единую систему идентификации и аутентификации (ЕСИА) для регистрации абитуриентов. После выпуска приказов о зачислении, поступившим проставляется статус «зачислен», они становятся видны на факультетах и в институтах, куда они поступили. После формирования деканатами студенческих групп студенты-первокурсники получают все возможности по работе с ЭИОС. Необходимо отметить, что ИС «Абитуриент», в свою очередь, интегрирована с ФИС ЕГЭ.

ИС «Парус» используется управлением кадров и бухгалтерией университета. С ними осуществляется интеграция на уровне данных: каждую ночь запускаются скрипты, обеспечивающие синхронизацию информации о структуре вуза и сотрудниках с ИС управления кадров, а также данных об оплате студентами обучения с ИС бухгалтерии.

Помимо прочего, в ЭИОС загружены данные Управления режима, безопасности и гражданской обороны о пропусках на автомобили сотрудников и студентов. Теперь дежурные на КПП имеют возможность оперативно работать с этой информацией.

Через размещенные в ЭИОС ссылки на литературу осуществляется интеграция с внешними ЭБС, с которыми у ПГУ имеются договора о сотрудничестве. Также широко используются ссылки из ЭИОС на размещенные в электронном каталоге научной библиотеки карточки печатных изданий.

Портал ПГУ тесно интегрирован с ЭИОС. Из ЭИОС извлекается представленная на портале структура подразделений ПГУ. Также на портале публикуются обязательные к размещению в открытом доступе документы (аннотации и рабочие программы дисциплин), находящиеся в ЭИОС.

Полный охват всех сотрудников и студентов вуза учетными записями в ЕСИА ЭИОС породил массу дополнительных возможностей. Так, уже сейчас по логину и паролю в ЭИОС осуществляется доступ в wi-fi на территории университета.

В качестве дополнительного функционала в составе ЭИОС реализован модуль для учета почасовой нагрузки, который не только автоматизирует работу планового отдела, но и обеспечивает кафедры данными для распределения учебных занятий для почасовиков.

В ближайших планах – не только совершенствование базового функционала ЭИОС, но и расширение интеграции с другими используемыми ИС. Первоочередной задачей является интеграция с ЭДО «Директум». В настоящее время в пилотном режиме возможна подготовка проектов переводных приказов для студентов. Далее эти проекты в формате doc загружаются в «Директум». После всех согласований и подписания ректором в ЭИОС заносятся выходные данные приказа, ему выставляется статус «исполнен», после чего в ЭИОС выполняются все вытекающие из приказа действия (студенты переводятся на следующий курс, им становятся доступны УМК дисциплин, изучаемых на этом курсе и т.д.). В ближайших планах – формирование проектов всех приказов, связанных с движением контингента студентов.

На базе ЭИОС разрабатываются дополнительные модули, предназначенные для автоматизации воинского учета в вузе, поддержки службы трудоустройства студентов, обеспечения воспитательного процесса.

Таким образом, разработанная ЭИОС университета стала основой его дальнейшей информатизации, направленной как на совершенствование информатизации собственно учебного процесса (расширение состава подготавливаемых форм отчетов, совершенствование юзабилити), так и на проникновение информатизации в другие сферы деятельности вуза.

Библиографический список

1. Мещеряков, В. А. Электронная информационно-образовательная среда университета в эру ФГОС ВО / В. А. Мещеряков, А. В. Антонов, А. М. Бершадский, И. Г. Кревский // Научно-образовательная информационная среда XXI века : материалы X Всерос. науч.-практ. конф. (20–23 сентября 2016 г.). – Петрозаводск, 2016. – С. 108–111.

А. В. Шмид

Нациолнальный университет «Высшая школа экономики (МИЭМ)»
ЗАО «ЕС-лизинг», Москва, Россия

ЦИФРОВАЯ ЭКОНОМИКА – ВАШЕ БУДЕЩЕЕ. ЧТО ЭТО ТАКОЕ И КАК НАЙТИ В НЕЙ СВОЕ МЕСТО



Историческое здание НИЦЭВТ – центр реализации программы ЕС ЭВМ: преодоления стратегического отставания СССР в создании АСУ



Назад в Будущее!

2

IBM – НИЦЭВТ. История взаимоотношений

Александр Шмид начал работать в НИЦЭВТе в 1969 году

- 1988 – Главный конструктор по CAD ЭВМ стран СЭВ
- 1989 – Главный конструктор Совместной (IBM – Минрадиопром СССР) программы в области больших ЭВМ ЕС
- До января 2004г.- Главный конструктор НИЦЭВТ
- 1994 – ЗАО «ЕС-лизинг»

Январь1991г., США

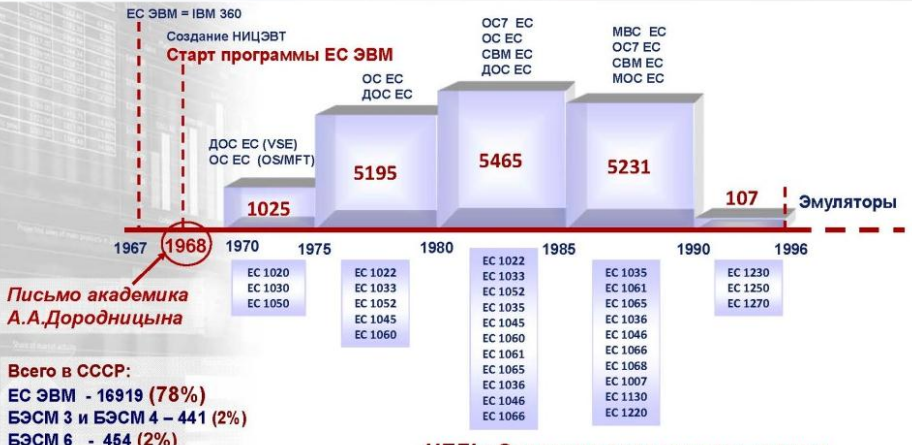


Министр МРП СССР В.И Шишко

А.В. Шмид

3

Количественные показатели программы ЕС ЭВМ



Всего в СССР:
ЕС ЭВМ - 16919 (78%)
БЭСМ 3 и БЭСМ 4 – 441 (2%)
БЭСМ 6 - 454 (2%)
МИНСК 22,23,32 - 3906 (18%)

ЦЕЛЬ: **Создание средств повышения производительности труда программистов**

www.COMPUTER-MUSEUM.ru

4

ЕС-лизинг

Содержание

1. Big Data: Революция в области философии и технологий принятия корпоративных решений
2. Интеллектуальность решающего центра. Стратегические угрозы XXI века в область ИТ: борьба за качество корпоративных решений – компьютеры против людей
3. Информированность решающего центра. Амнезия корпораций
4. Потоковая обработка (streaming): преодоление проклятия размерности при хранении данных
5. Цифровая экономика как следствие цифровой революции
6. Макроуровень стратегии перехода РФ к цифровой экономике: обеспечение численности работающих и производительности труда
7. Производительность труда: промышленные и домашние технологии - мировые лидеры коммерческих платформ. О роли OPEN SOURCE.
8. Обеспечение численности работающих. Концепция подготовки кадров цифровой экономики на примерах: от деталировки к созданию (сборке) и обучению ЭКСПЕРТНЫХ СИСТЕМ.



5

ЕС-лизинг

«Данные — это новая нефть»
британский математик Клайв Хамби 2006г.
Торговая сеть Tesco-программа лояльности

«В мире самый ценный ресурс уже не нефть, а данные»
Журнал «The Economist» 2017г.

Big Data: Революция в области философии и технологий принятия корпоративных решений



"В ближайшие 5 лет все компании на рынке разделятся на победителей и побежденных в зависимости от качества корпоративных решений.
И опыт, и интуиция не помогут. Людей надо переучивать. Нигде этому сейчас не учат, но мы будем!"

Вирджиния Рометти – IBM CEO
11 марта 2013 года

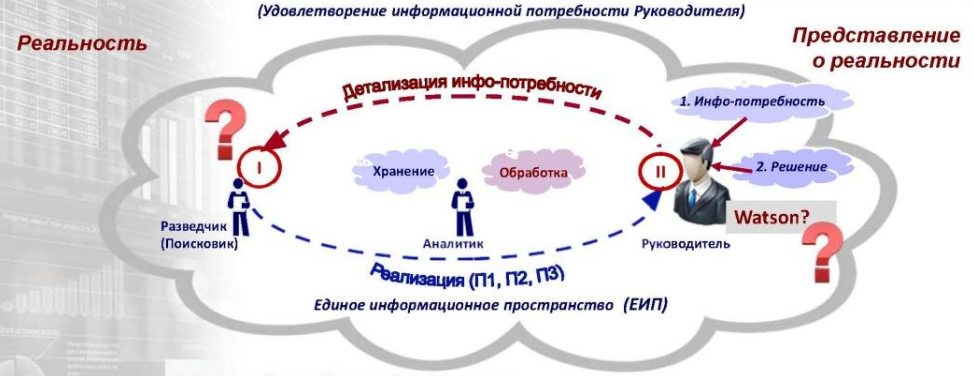
6

ЕС-лизинг

Традиционная постановка задачи информационной поддержки принятия решений

Информированность и интеллектуальность решающего центра

(Удовлетворение информационной потребности Руководителя)



Инфо потребность должна удовлетворяться:

БЫСТРО – информация об изменениях обстановки должна поступать к моменту принятия решения (быстрее, чем у конкурентов)

ПОЛНО – необходимая и достаточная для принятия решений

ДОСТОВЕРНО – исключение фальсификации

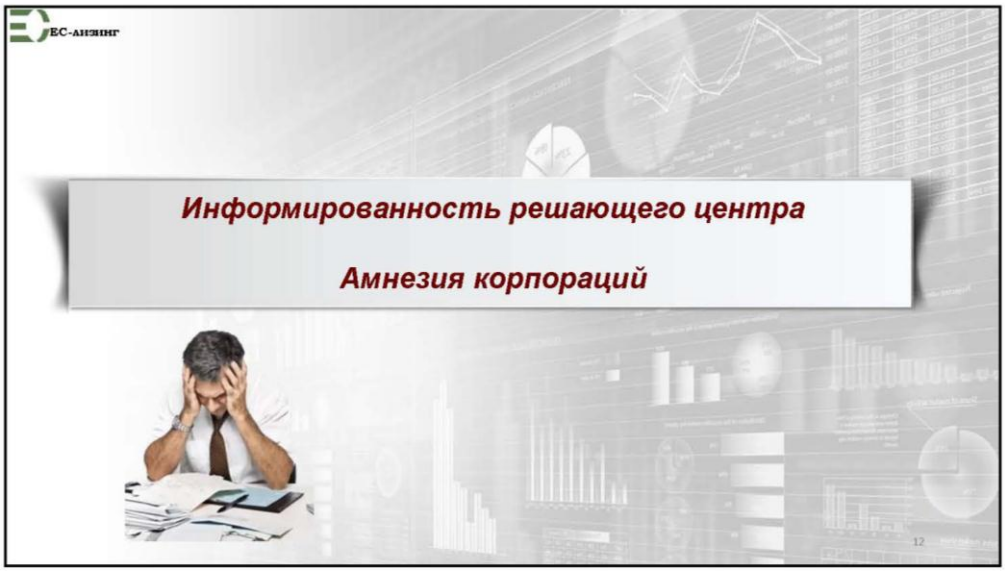
7


ЕС-лизинг
Традиционная постановка задачи информационной поддержки принятия решений и условия ее успешности
Историческая схема принятия решений
(Удовлетворение информационной потребности Руководителя)
Реальность
Представление о реальности
Детализация инфо-потребности
1. Инфо-потребность
2. Решение
Хранение
Обработка
I
II
Разведчик (Поисковик)
Аналитик
Руководитель
Реализация (П1, П2, П3)
Единое информационное пространство (ЕИП)
Инфо потребность должна удовлетворяться:
БЫСТРО – информация об изменениях обстановки должна поступать к моменту принятия решения (быстрее, чем у конкурентов)
ПОЛНО – необходимая и достаточная для принятия решений
ДОСТОВЕРНО – исключение фальсификации
Две координаты успешности
Интеллектуальность
Информированность
0
1
2
3
Зона успешных решений
1. «Неосведомленный гений»
2. «Информированный дурак»
3. Успешный руководитель
8

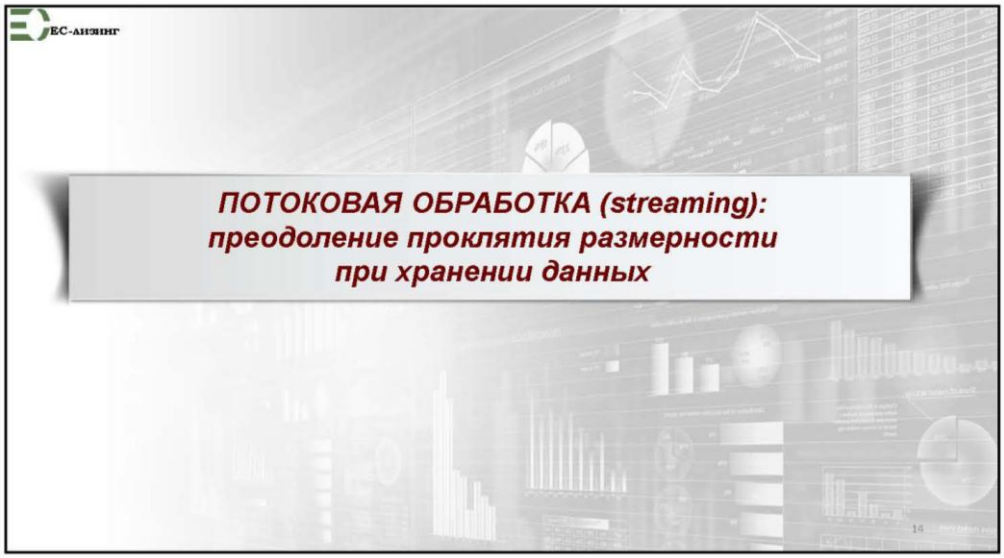

ЕС-лизинг
Интеллектуальность решающего центра
Стратегические угрозы XXI века в область ИТ:
борьба за качество корпоративных решений –
компьютеры против людей
9


ЕС-лизинг
Стратегический прогноз IBM развития ИТ 2005 – 2015гг.
и его реализация
Цюрих, ноябрь 2005г.
Лаборатория стратегического планирования IBM
К 2015 году компьютеры будут принимать решения ЛУЧШЕ людей во многих областях бизнеса

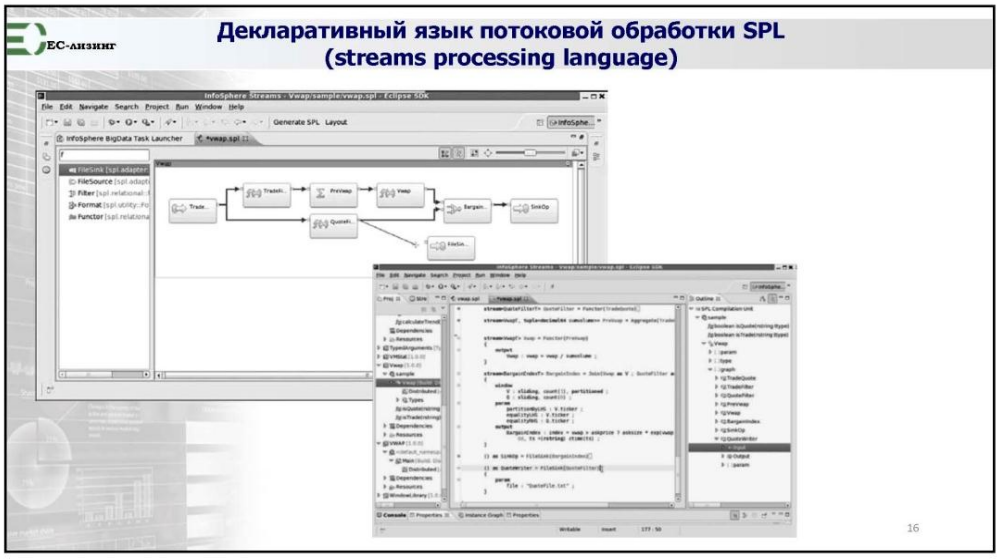

ЕС-лизинг
Цифровая революция: Компьютеры против людей
Дорожная карта реализации стратегического прогноза 2005-2015гг
IBM
WATSON
История развития IBM Watson
IBM Research Project (2006–)
Jeopardy! Grand Challenge (Feb 2011)
Watson for Healthcare (Aug 2011–)
Watson for Financial Services (Mar 2012–)
Watson Industry Solutions (2012–)
R&D
Demonstration
Commercialization
Expansion
Cross-industry Applications
http://www.dailytechinfo.org/infotech/2926-ocherednaya-pobeda-superkompyutera-watson-i-polnoe-porazhenie-medikov-kardiologov.html
2016г. Консилиум: Watson против ДВУХ бригад кардиологов США
Watson победил
Шаги реализации прогноза:
2011 год – Watson победитель игры Jeopardy (USA) - прототип телепрограммы «Своя игра»
Лучше людей!
2013 год – ошибки в диагнозе по онкологии:
40% - люди
3-5% - Watson
IBM
WATSON
Помощь онкологам в диагностировании и выборе схемы лечения рака
20-44%

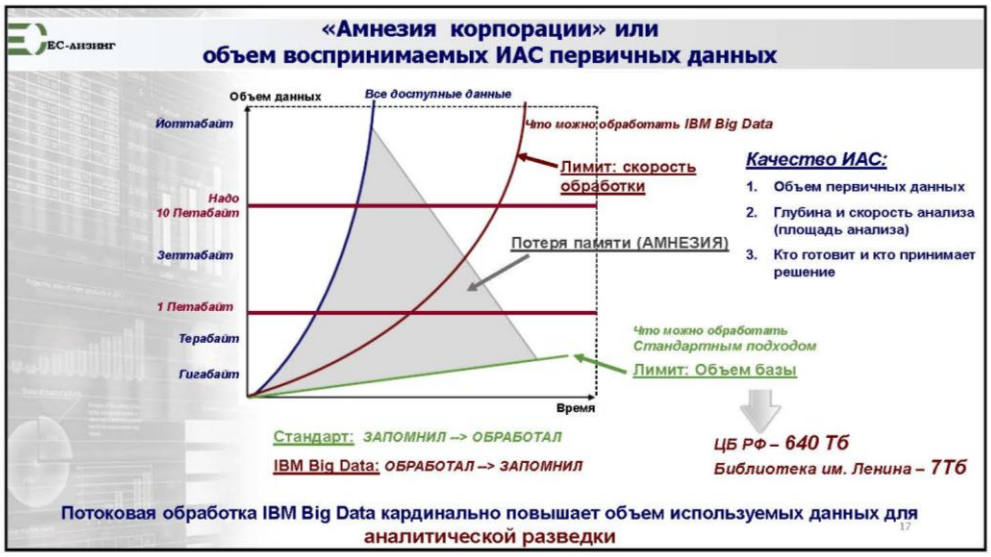

ЕС-лизинг
Информированность решающего центра
Амнезия корпораций


ЕС-лизинг
Проклятие размерности. Сколько информации мы теряем?
Figure 2
Information Creation and Available Storage
Exabytes
1,800
1,600
1,400
1,200
1,000
0,800
0,600
0,400
0,200
Information Created
Available Storage, 2007
Tape 21%
Optical 22%
Other 1%
Disk 56%
264 EB
Available Storage
2005
2006
2007
2008
2009
2010
2011
Source: IDC, 2008
IDC, 2008
Рометти: «В 2013 году 2 дня = всему 2003 (рост в 178 раз!)»


ЕС-лизинг
ПОТОКОВАЯ ОБРАБОТКА (streaming):
преодоление проклятия размерности
при хранении данных


ЕС-лизинг
2002г.
IBM получает заказ на разработку технологии:
1. Любые виды источников
2. Любая скорость потока данных от источника
3. Принятие решения – в потоке (миллисекунды)
15


ЕС-лизинг
Декларативный язык потоковой обработки SPL
(streams processing language)
16

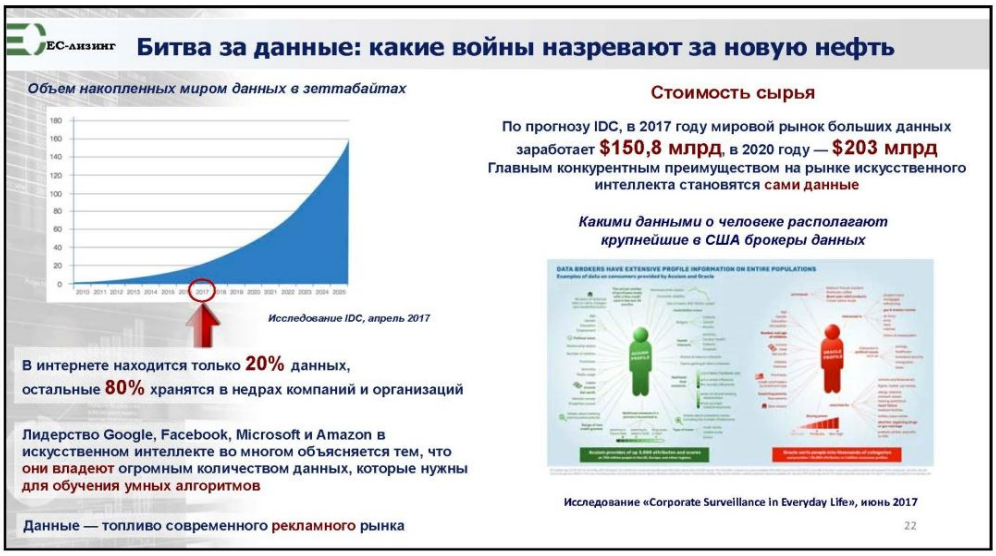

«Амнезия корпорации» или
объем воспринимаемых ИАС первичных данных
Объем данных
Все доступные данные
Йоттабайт
Что можно обработать IBM Big Data
Лимит: скорость обработки
Качество ИАС:
1. Объем первичных данных
2. Глубина и скорость анализа (площадь анализа)
3. Кто готовит и кто принимает решение
Надо 10 Петабайт
Потеря памяти (АМНЕЗИЯ)
Зеттабайт
1 Петабайт
Терабайт
Что можно обработать Стандартным подходом
Гигабайт
Лимит: Объем базы
Время
Стандарт: ЗАПОМНИЛ --> ОБРАБОТАЛ
IBM Big Data: ОБРАБОТАЛ --> ЗАПОМНИЛ
ЦБ РФ – 640 Тб
Библиотека им. Ленина – 7Тб
Потоковая обработка IBM Big Data кардинально повышает объем используемых данных для аналитической разведки

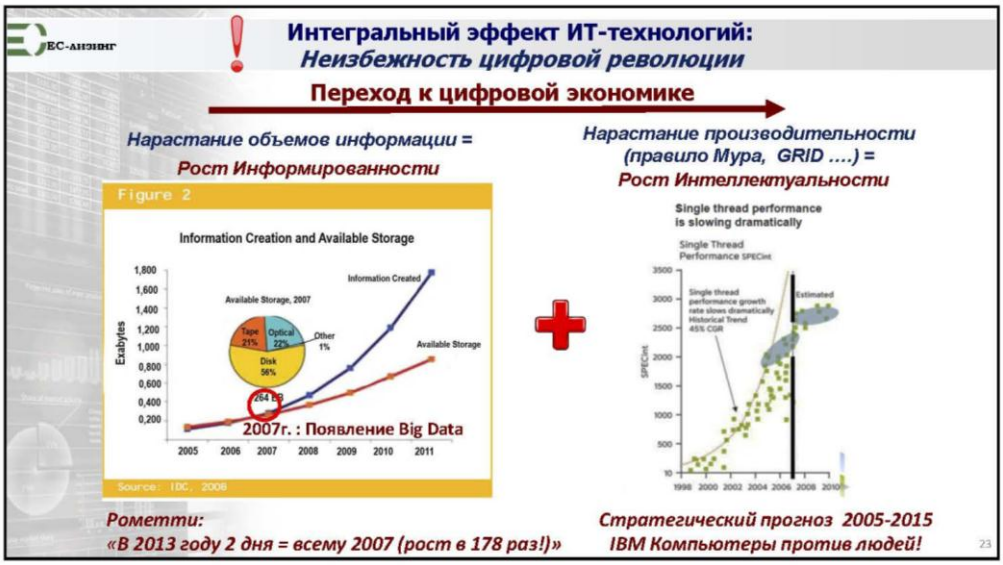

Большие данные – горячая тема, потому что технологии сделали возможным анализ ВСЕХ доступных данных
Эффективно с точки зрения затрат управлять и анализировать
все доступные данные,
в их первозданном виде – структурированные,
неструктурированные, потоковые
Website
Social Media
facebook
Billing
ERP
CRM
RFID
Network Switches

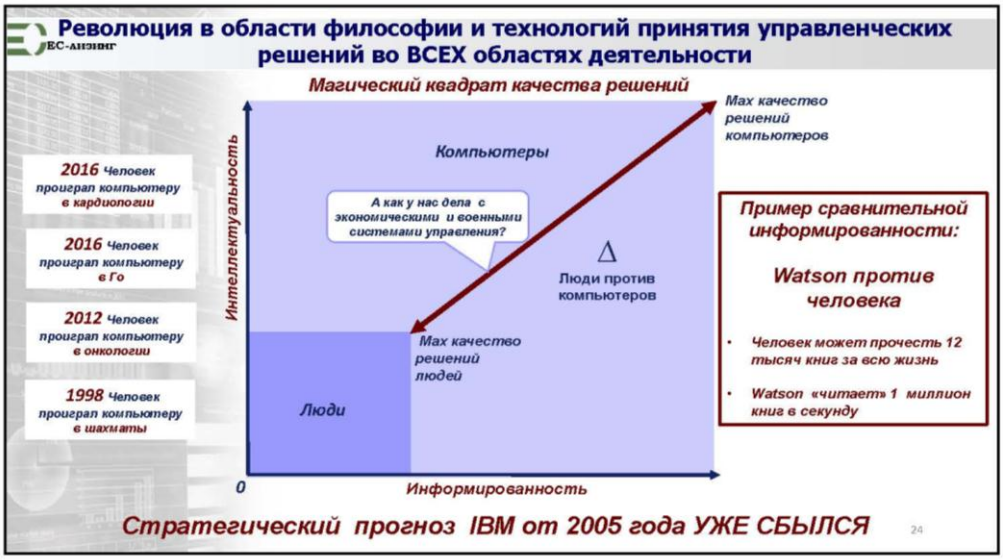

Видеонадзор: безопасность или тотальный контроль?
Хронология событий
1942
1949
1960
1973
2005
2012
2015
2017
2019
2020
2025
В Германии компанией Siemens создана первая в мире система видеонаблюдения за испытаниями ракет «Фау-2» на полигоне Пенемюнде.
Опубликован «1984» Джорджа Оруэлла
Полиция Лондона устанавливает две временные камеры на Трафальгарской площади для мониторинга за толпой, пришедшей наблюдать за официальным визитом тайской королевской семьи.
На Таймс-сквер появляется первая постоянная стационарная камера теленаблюдения
В Великобритании введена национальная система автоматического распознавания автомобильных номеров
Разработаны приложения к детским смартфонам для определения местоположения и фотосъемки
«Черные ящики» в США становятся обязательными для всех новых автомобилей
Детекторы злонамеренности внедрены почти во всех общественных местах
Миллиарды видеокамер — повсюду
Правительства располагают образцами ДНК всех до единого жителей планеты
Ричард Уотсон
«Будущее. 50 идей, о которых нужно знать.»


ЕС-лизинг
Видеонадзор: безопасность или тотальный контроль?
В Великобритании установлено более 4 200 000 камер скрытого видеонаблюдения (CCTV) — по одной на 14 граждан. Средний британец ежедневно появляется на экранах видеонаблюдения до 300 раз.
Правительство Индии запустило 10-летний проект создания самой обширной в мире идентификационной базы данных.
Консультативная группа высокого уровня по будущему европейской политики в сфере внутренних дел после 2010 года («Группа будущего») выпустила документ, в котором рекомендует наблюдение за «каждым объектом в пользовании индивидуума, каждым банковским переводом и почти каждым перемещением».
Это может включать в себя налоговые платежи, историю найма, банковские реквизиты и историю пользования банковскими картами, сведения о здоровье, информацию о перемещениях и даже участии в социальных сетях.
В 2002 году Пентагон предложило проект TIA (Total Information Awareness, «Полная информационная прозрачность») - создание компьютерных средств и программ, способных анализировать многочисленные базы данных (в том числе содержащие приватную информацию о людях) и предсказывать террористические атаки, а также выявлять потенциальных террористов. (в 2003 году Конгресс закрыл пентагоновский проект.)
20


ЕС-лизинг
«ВКонтакте» под прицелом спецслужб США: кто уже на крючке?
SocioSpyder ежедневно отслеживает и анализирует активность миллиарда пользователей сети Facebook и полмиллиарда сообщений Twitter.
SOCIOSPYDER
Теперь очередь за российским сегментом сети «Вконтакте». Согласно каталогу пользователей, в ней зарегистрировано уже 443 миллиона аккаунтов.
Есть, где разгуляться американским аналитикам.
Чем это грозит пользователям социальных сетей?
Все комментарии, картинки, тексты, твиты, видео, запросы и прочая информация о вас, будет скопирована, загружена на хранение и проанализирована аналитиками из американских спецслужб.
Никто из пользователей даже не узнает, что оказался «под колпаком», конечно же, до поры до времени. Зато при необходимости каждого можно будет подловить «на крючок». С какой целью? Вот тогда и узнаете...
Facebook 1 млрд
Twitter 0,5 млрд
В контакте 443 млн
21


ЕС-лизинг
Битва за данные: какие войны назревают за новую нефть
Объем накопленных миром данных в зеттабайтах
Исследование IDC, апрель 2017
В интернете находится только 20% данных, остальные 80% хранятся в недрах компаний и организаций
Лидерство Google, Facebook, Microsoft и Amazon в искусственном интеллекте во многом объясняется тем, что они владеют огромным количеством данных, которые нужны для обучения умных алгоритмов
Данные — топливо современного рекламного рынка
Стоимость сырья
По прогнозу IDC, в 2017 году мировой рынок больших данных заработает $150,8 млрд, в 2020 году — $203 млрд
Главным конкурентным преимуществом на рынке искусственного интеллекта становятся сами данные
Какими данными о человеке располагают крупнейшие в США брокеры данных
DATA BROKERS HAVE EXTENSIVE PROFILE INFORMATION ON ENTIRE POPULATIONS
Исследование «Corporate Surveillance in Everyday Life», июнь 2017
22


ЕС-анэинг
Интегральный эффект ИТ-технологий:
Неизбежность цифровой революции
Переход к цифровой экономике
Нарастание объемов информации =
Рост Информированности
Нарастание производительности
(правило Мура, GRID ....) =
Рост Интеллектуальности
Figure 2
Information Creation and Available Storage
Exabytes
Information Created
Available Storage
Available Storage, 2007
Tape 21%
Optical 22%
Other 1%
Disk 56%
264 EB
2007г. : Появление Big Data
Source: IDC, 2008
Single thread performance is slowing dramatically
Single Thread Performance SPECint
Single thread performance growth rate slows dramatically Historical Trend 45% CGR
Estimated
Рометти:
«В 2013 году 2 дня = всему 2007 (рост в 178 раз!)»
Стратегический прогноз 2005-2015
IBM Компьютеры против людей!

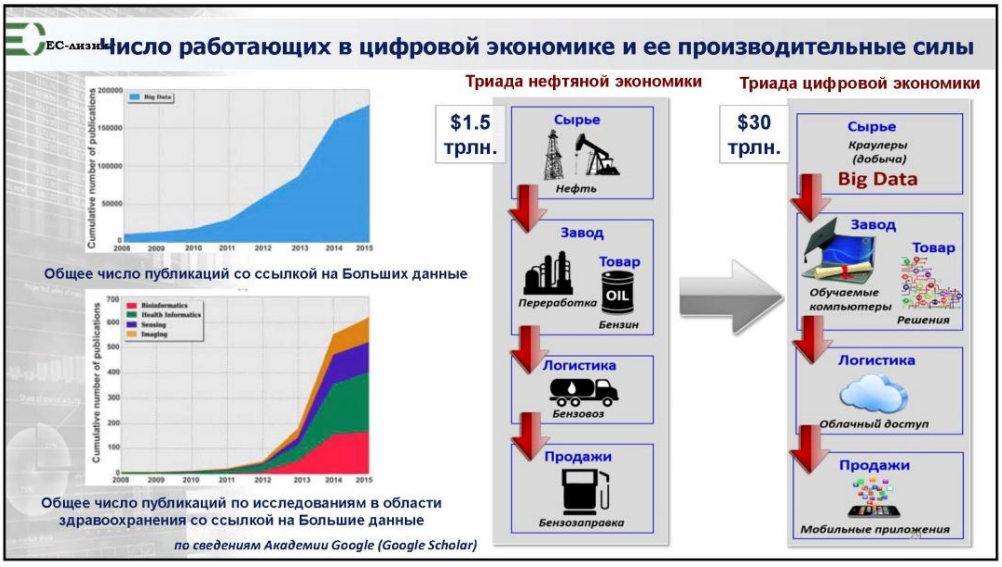

ЕС-анэинг
Революция в области философии и технологий принятия управленческих решений во ВСЕХ областях деятельности
Магический квадрат качества решений
2016 Человек проиграл компьютеру в кардиологии
2016 Человек проиграл компьютеру в Го
2012 Человек проиграл компьютеру в онкологии
1998 Человек проиграл компьютеру в шахматы
Интеллектуальность
Компьютеры
Max качество решений компьютеров
А как у нас дела с экономическими и военными системами управления?
Δ
Люди против компьютеров
Max качество решений людей
Люди
0
Информированность
Пример сравнительной информированности:
Watson против человека
Человек может прочесть 12 тысяч книг за всю жизнь
Watson «читает» 1 миллион книг в секунду
Стратегический прогноз IBM от 2005 года УЖЕ СБЫЛСЯ


ЕС-анэинг
Big Data: Изменение парадигмы при работе с данными
Традиционный подход
Структурный и повторяемый анализ
«Запомнил - обработал»
Бизнес
Определяет: что спросить
ИТ
Структурирует данные для ответа на вопрос
Месячная отчетность
Анализ прибыльности
Анализ анкет
Ограничение: память
Большие данные
Итеративность и исследование
«Обработал – запомнил»
ИТ
Обеспечивает платформу для креативного анализа (креативная команда on-line)
Бизнес
Исследует: что можно спросить
Отношение к бренду
Стратегия продуктов
Оптимизация ресурсов
Пример IBM - MPП


ЕС-лизинг
Современная схема принятия решений:
Заявка на НОВОЕ (срочно!) образование
РЕШЕНИЕ:
Меры воздействия на объект управления
CEO
Подтверждение решения
Задания интегральной инфо потребности
Детализация области интересов
Область интересов
Обсуждение области интересов
Навигация
Поиск
Визуализация
Фиксация
Интеграция
Обучение
Пробное решение
Watson Type
Принятие решения
Цифровая реальность
Поиск в области интересов
Креативная команда
Обучение в области интересов
Кардинальное изменение исторической ролевой модели, требований к персоналу, структуры бизнес-процессов и технологий подготовки и принятия решений

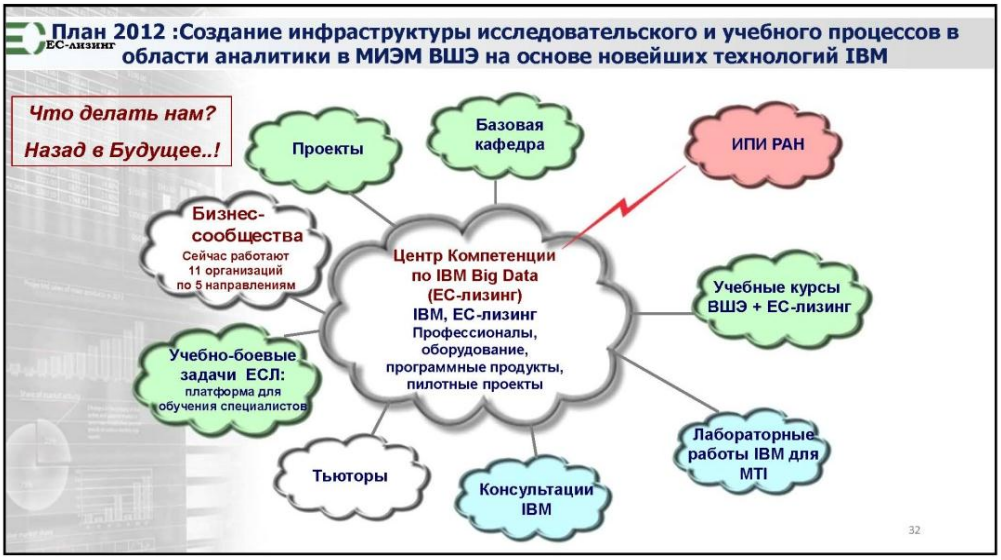

ЕС-лизинг
Цифровая экономика как следствие цифровой революции
27


ЕС-лизинг
Что такое любая экономика?
Количество работающих
X
Производительность труда
2012 год:
производительность труда на одного работающего
McDonald's $ 40 000
Wal-Mart $150 000
Microsoft $500 000
Google $1 000 000
$30 трлн.
$300 млрд.
$3 млрд.
2012
2016
2025
Два вопроса:
1. Как повысить производительность труда?
2. Как обеспечить численность работающих? Где взять столько работающих?
Число работающих в цифровой экономике
3 000 чел.
30 млн. чел.
2012?
2025?
28

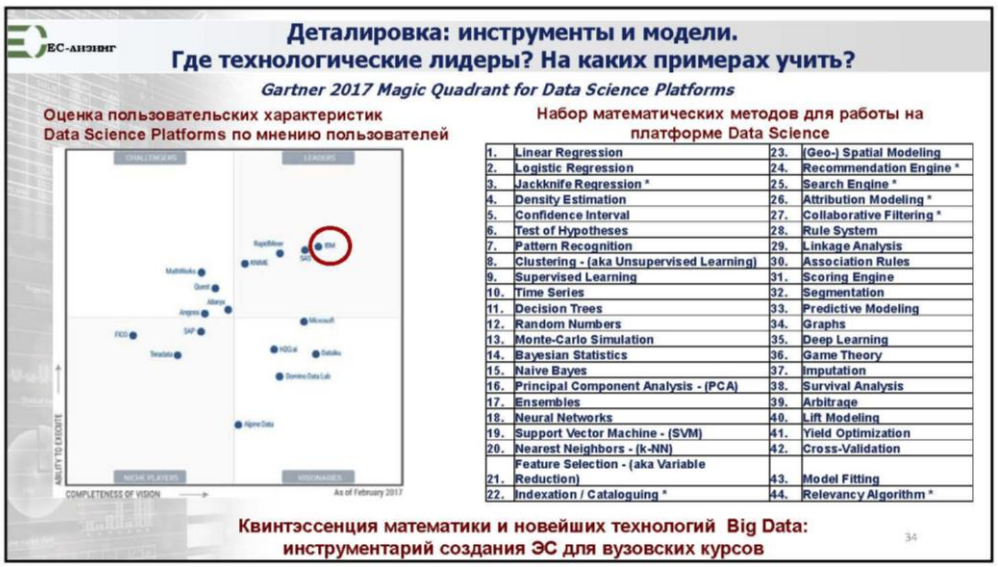

ЕС-лизинг
Число работающих в цифровой экономике и ее производительные силы
Big Data
Cumulative number of publications
Общее число публикаций со ссылкой на Больших данные
Bioinformatics
Health Informatics
Sensing
Imaging
Общее число публикаций по исследованиям в области здравоохранения со ссылкой на Большие данные
по сведениям Академии Google (Google Scholar)
Триада нефтяной экономики
$1.5 трлн.
Сырье
Нефть
Завод
Переработка
Товар
OIL
Бензин
Логистика
Бензовоз
Продажи
Бензозаправка
Триада цифровой экономики
$30 трлн.
Сырье
Краулеры (добыча)
Big Data
Завод
Обучаемые компьютеры
Товар
Решения
Логистика
Облачный доступ
Продажи
Мобильные приложения


ЕС-лизинг
Как выглядит цифровая экономика и как нам занять достойное место?
Прогноз института McKinsey - 2025
A gallery of disruptive technologies
Estimated potential economic impact of technologies across sized applications in 2025, $ trillion, annual
0 1 2 3 4 5 6 7 8 9 10 11
1. Mobile Internet
2. Automation of knowledge work
3. Internet of Things
4. Cloud
5. Advanced robotics
6. Autonomous and near-autonomous vehicles
7. Next-generation genomics
8. Energy storage
9. 3-D printing
10. Advanced mterials
11. Advanced oil and gas exploration and recovery
12. Renewable energy
Экономика знаний
$30 трлн.
Нефтегазовый комплекс
$1.5 трлн.
SOURCE: McKinsey Global Institute
30


ЕС-лизинг
МАКРОУРОВЕНЬ СТРАТЕГИИ ПЕРЕХОДА РФ к ЦИФРОВОЙ ЭКОНОМИКЕ: обеспечение ЧИСЛЕННОСТИ РАБОТАЮЩИХ и ПРОИЗВОДИТЕЛЬНОСТИ ТРУДА
(назад, в будущее!)
31


План 2012 :Создание инфраструктуры исследовательского и учебного процессов в области аналитики в МИЭМ ВШЭ на основе новейших технологий IBM
ЕС-лизинг
Что делать нам?
Назад в Будущее..!
Проекты
Базовая кафедра
ИПИ РАН
Бизнес-сообщества
Сейчас работают 11 организаций по 5 направлениям
Центр Компетенции по IBM Big Data (ЕС-лизинг)
IBM, ЕС-лизинг
Профессионалы, оборудование, программные продукты, пилотные проекты
Учебные курсы ВШЭ + ЕС-лизинг
Учебно-боевые задачи ЕСЛ:
платформа для обучения специалистов
Лабораторные работы IBM для MTI
Тьюторы
Консультации IBM
32


ЕС-лизинг
ПРОИЗВОДИТЕЛЬНОСТЬ ТРУДА:
промышленные и домашние технологии -
мировые лидеры коммерческих платформ.
О роли OPEN SOURCE
33


ЕС-лизинг
Деталировка: инструменты и модели.
Где технологические лидеры? На каких примерах учить?
Gartner 2017 Magic Quadrant for Data Science Platforms
Оценка пользовательских характеристик Data Science Platforms по мнению пользователей
Набор математических методов для работы на платформе Data Science
1. Linear Regression
2. Logistic Regression
3. Jackknife Regression *
4. Density Estimation
5. Confidence Interval
6. Test of Hypotheses
7. Pattern Recognition
8. Clustering - (aka Unsupervised Learning)
9. Supervised Learning
10. Time Series
11. Decision Trees
12. Random Numbers
13. Monte-Carlo Simulation
14. Bayesian Statistics
15. Naive Bayes
16. Principal Component Analysis - (PCA)
17. Ensembles
18. Neural Networks
19. Support Vector Machine - (SVM)
20. Nearest Neighbors - (k-NN)
21. Feature Selection - (aka Variable Reduction)
22. Indexation / Cataloguing *
23. (Geo-) Spatial Modeling
24. Recommendation Engine *
25. Search Engine *
26. Attribution Modeling *
27. Collaborative Filtering *
28. Rule System
29. Linkage Analysis
30. Association Rules
31. Scoring Engine
32. Segmentation
33. Predictive Modeling
34. Graphs
35. Deep Learning
36. Game Theory
37. Imputation
38. Survival Analysis
39. Arbitrage
40. Lift Modeling
41. Yield Optimization
42. Cross-Validation
43. Model Fitting
44. Relevancy Algorithm *
Квинтэссенция математики и новейших технологий Big Data:
инструментарий создания ЭС для вузовских курсов
34


ЕС-анализ
Лидеры домашних платформ
Gartner Magic Quadrant for Business Intelligence (BI) 2017
Gartner Magic Quadrant for Business Intelligence and Analytics Platforms
Trend 2016 vs 2017
Benjamin Arnulf
CHALLENGERS
LEADERS
NICHE PLAYERS
VISIONARIES
ABILITY TO EXECUTE
COMPLETENESS OF VISION
As of February 2017
Tableau
Microsoft
Qlik
SAP
SAS
IBM
Oracle
Salesforce
Zoomdata
Pentaho
Pyramid Analytics
Datameer


ЕС-анализ
Требования к функциям платформы Больших Данных
Поиск и навигация источников данных в киберпространстве
InfoSphere Data Explorer и т.д.
Подключение источников и анализ данных «в покое»
Hadoop File System
IBM Content Analyzer
Подключение источников и анализ данных «в движении»
InfoSphere Streams и т.д.
Традиционные функции работы со структурированными данными
Netezza и т.д.
Интеграция всех видов данных для комплексного анализа
IBM Information Server
IBM Change Data Capture
Автоматизация принятия решений и построение гипотез и прогнозов
IBM Cognos
IBM SPSS
36

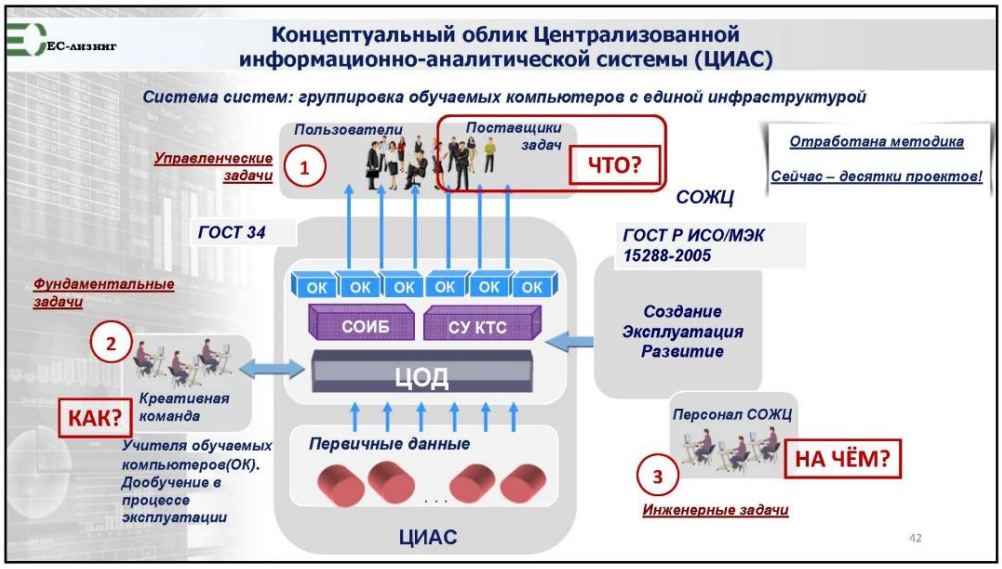

ЕС-анализ
Обеспечение ЧИСЛЕННОСТИ РАБОТАЮЩИХ.
Концепция подготовки кадров ЦИФРОВОЙ ЭКОНОМИКИ на примерах: от деталировки к созданию (сборке) и обучению ЭКСПЕРТНЫХ СИСТЕМ
37

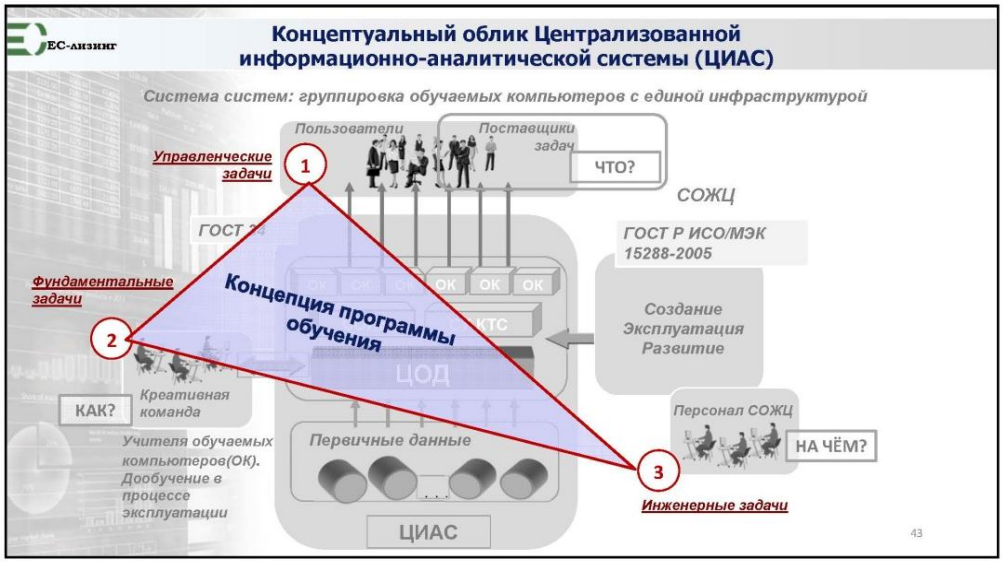

ЕС-лизинг
Сборка: система в целом - ЦИАС АПК РФ
Чему учить (пример): Стратегическое управление состоянием АПК.
Три режима: 1. Мониторинг : Сопоставление плана и факта
2. Анализ: Разработка ВАРИАНТОВ мер воздействия на АПК
3. Прогнозирование: Влияние мер воздействия на состояние будущего факта (предсказание последствий мер воздействия)
Состояния АПК: Плановое, Фактическое, Прогнозируемое
Фактор «ЗА» - Меры воздействия
Мониторинг факта
Факт
План
Прогноз
Фактор «ПРОТИВ» - риски АПК
План
Факт
Прогноз
t Время
2016 2017 2018 2019 2020
Меры воздействия МСХ
на людей, с/х предприятия, банки, рынки, страховые компании
1. Субсидии
2. Интервенции
3. Налогообложение
4. Таможенно-тарифное регулирование
Макроэкономическая (отсроченная) реакция АПК !
Риски:
1. макроэкономические риски
2. внешнеторговые риски
3. природные риски
Big Data !
38


ЕС-лизинг
ПЛАН РЕАЛИЗАЦИИ: мониторинг, анализ и прогнозирование на основе математических моделей
(Из Конкурсной документации ЦИАС СГИО СХ)
Ключевые аргументы
Объекты анализа и прогноза
Результаты анализа и прогноза
1 Производственные мощности: поля, техника, поголовье и др.
2 Финансы: Кредитоспособность; ликвидность; платежеспособность; рентабельность
3 Меры воздействия: Субсидии; интервенции; таможенно-тарифное регулирование; налогообложение
4 Риски: макроэкономические риски; внешнеторговые риски; природные риски
Математические модели
РФ
Округ (8)
Регион (85)
Район (2300)
СХП (1000000)
5 6 7 8 9
Прогнозные
-Производственная устойчивость;
-Финансовая устойчивость;
и др.
Отчетные
-Показатели Госпрограммы;
-Показатели Продовольственной безопасности;
и др.
39


ЕС-лизинг
Трудоемкость создания и источники данных ЦИАС СГИО СХ
ЦИАС СГИО СХ
ППК ПУ
ППК ВПР
ППК Геосервис
ППК ИС ПК ГП
1,5 - 2%
Инфраструктура
НСИ
ИС ПК ГП
ССДД
Внешнеэкономические Риски
РОССТАТ
Макроэкономические Риски
Природно-климатические Риски
СХТП
40


ЕС-лизинг
Схема аналитической системы (ЦИАС БР) для применения в АПК
Сочетание фундаментальных, инженерных и управленческих задач
ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ к ЦИАС следуют из состава задач, решаемых ЛЮДЬМИ
Что делают люди
Чем поможет ЦИАС
ЦИАС БР – 5 лет
МСХ – 90 дней
Задачи стратегического управления
Отчетность по системе стратегического управления (план/ факт), корректировка ГП
Отчетность межведомственная (42 ГП ведомств по АПК)
Управление бюджетом
Разработка мер воздействия
Вспомогательные задачи по управлению
Прогноз состояния АПК
Оценка рисков
Мониторинг цен
Анализ мер воздействия
Аналитика по конкуренции
Анализ компенсационных мер
Требуемая функциональность СГИО СХ
СГИО СХ
~ 40 новых задач в ИАС
Инфраструктура ЦИАС
ИАП
ЦОД
СОЖЦ + СОИБ
1
Управленческие задачи
Фундаментальные задачи
2
- Новые приложения
- Новая модель
- Требования к данным
3
Инженерные задачи
41


ЕС-лизинг
Концептуальный облик Централизованной информационно-аналитической системы (ЦИАС)
Система систем: группировка обучаемых компьютеров с единой инфраструктурой
Пользователи
Поставщики задач
ЧТО?
Управленческие задачи
1
Отработана методика
Сейчас – десятки проектов!
СОЖЦ
ГОСТ 34
ГОСТ Р ИСО/МЭК 15288-2005
Фундаментальные задачи
2
ОК
СОИБ
СУ КТС
ЦОД
Создание
Эксплуатация
Развитие
КАК?
Креативная команда
Учителя обучаемых компьютеров(ОК). Дообучение в процессе эксплуатации
Первичные данные
Персонал СОЖЦ
НА ЧЁМ?
3
Инженерные задачи
ЦИАС
42


ЕС-лизинг
Концептуальный облик Централизованной информационно-аналитической системы (ЦИАС)
Система систем: группировка обучаемых компьютеров с единой инфраструктурой
Пользователи
Поставщики задач
ЧТО?
Управленческие задачи
1
СОЖЦ
ГОСТ Р ИСО/МЭК 15288-2005
Фундаментальные задачи
Концепция программы обучения
2
ОК
КТС
ЦОД
Создание
Эксплуатация
Развитие
КАК?
Креативная команда
Учителя обучаемых компьютеров(ОК). Дообучение в процессе эксплуатации
Первичные данные
Персонал СОЖЦ
НА ЧЁМ?
3
Инженерные задачи
ЦИАС
43

Обучение деталировке в МИЭМ НИУ ВШЭ

Платформа IBM Big Data: Конструктор (акселератор) для сборки ЦИАС

Лабораторные работы IBM (MIT-USA) для освоения платформы в составе конструктора

В 2012 году в полном объеме в ЗАО «ЕС-лизинг»



Производительность труда при разработке ЭС в ЗАО «ЕС-лизинг» возросла ~ в 30 раз!

44

Состав и назначение лабораторных работ для платформы IBM Big Data

Что изучается	Используемые инструменты
1. Работа с неструктурированными данными в распределенной файловой системе Hadoop	Система Hadoop: • Распределенная файловая система HDFS • Параллельная обработка данных MapReduce
2. Управление процессами обработки неструктурированных данных с помощью BigInsights	
3. Анализ структурированных и неструктурированных данных с использованием BigSheets	IBM InfoSphere BigInsights BigSheets toolkit
4. Управление большими данными при помощи Big SQL	IBM InfoSphere BigInsights: Big SQL
5. Структурирование данных с помощью специализированных языков обработки текстовой информации	IBM InfoSphere BigInsights: AQL (Annotation Query Language)
6. Анализ потоковой информации с использованием специализированных языков обработки потоков данных	IBM InfoSphere Streams SPL (Streams Processing Language)
7. Управление обработкой потоковых данных	IBM InfoSphere Streams SPL (Streams Processing Language)
8. Исследование текстовой информации с помощью Watson Explorer	IBM InfoSphere Watson Explorer
9. Анализ структурированных и неструктурированных данных с помощью Watson Content Analytics	IBM Watson Content Analytics Studio
10. Выявление скрытых связей на основе анализа текстов с помощью i2	IBM i2

45

Платформа обучения магистрантов по сборке и применению ИАС

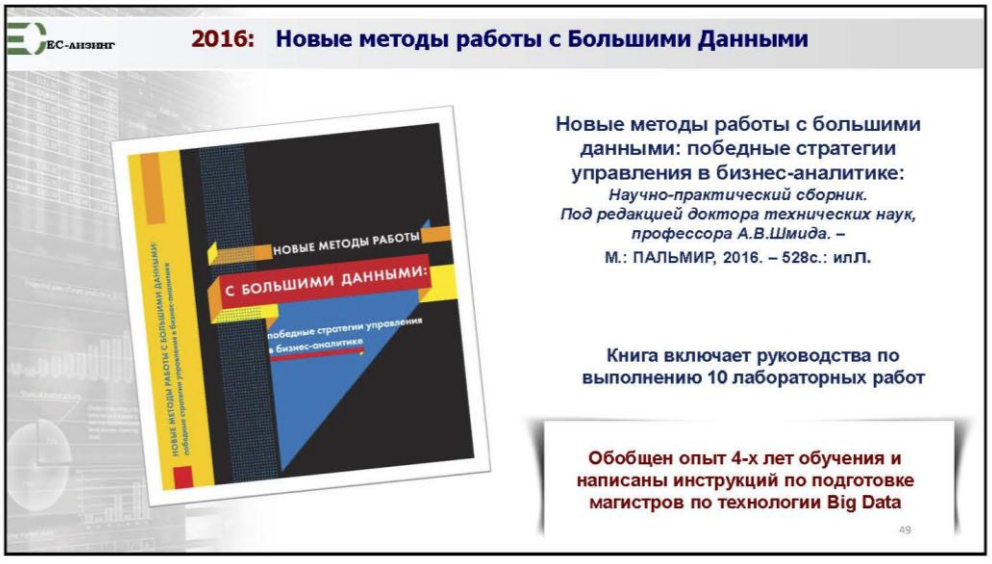
Учебно-боевые задачи 2013-2015гг базовой кафедры ИАС МИЭМ НИУ ВШЭ

№	Название задачи	Источники информации
1	Прогнозирование «бегства вкладчиков»	• Интернет-форумы • Средства массовой информации • Корпоративные источники информации • Финансовая отчетность • Документы управленческого учета • Социальные сети • Блоги • ЕГРЮЛ • Росстат • Арбитражные суды • exacom.ru + rusceo.com • rusprofile.ru • querycom.net • yell.ru • Машинные журналы • Сайты организаций • Интернет-магазины • и т.д.
2	Выявление аффилированных лиц	
3	Оценка степени удовлетворенности клиентов услугами	
4	Обнаружение фактов инсайда	
5	Изучение и анализ информационного поля в Интернет. Выделение троллинга.	
6	Поиск источников вброса, каналов и путей распространения информации.	
7	Дополнение информации о клиентах из открытых источников	
8	Прогнозирование оттока клиентов	
9	Выявление лиц, связанных с пропагандой и употреблением наркотиков	
10	Выявление организаций, осуществляющих деятельность на рынке микрофинансирования, не состоящие в Государственном реестре.	
11	Мониторинг и анализ цен конкурентов для формирования оптимальной цены и ассортимента	
12	Оценка и прогноз производственной устойчивости сельскохозяйственных товаропроизводителей	
13	Оценка финансовой устойчивости сельскохозяйственных товаропроизводителей	
14	Определение целевой аудитории для товаров парфюмерной компании среди участников социальных сетей	
15	Разработка инструментов и методов стандартизации адресов торговых точек	

46


ЕС-лизинг
Обязательные практики будущих магистров по проектированию ЭС
1-ый курс магистратуры: курсовой проект
2-ой курс магистратуры: дипломная работа по разработке ЭС
(практика проектирования в интересах заказчиков ЗАО «ЕС-лизинг»)
№ п/п
Тема междисциплинарной курсовой работы
1 Сравнительный анализ методов обработки больших данных и традиционных методов обработки данных
2 Исследование методов датификации процессов и явлений
3 Исследование цепочек создания ценности больших данных
4 Сравнительный анализ методов прогнозирования на основе больших данных
5 Анализ методов применения больших данных в экспертных системах
6 Анализ рисков использования больших данных
7 Исследование вопросов конфиденциальности при использовании больших данных
8 Прогнозирование поведения вкладчиков на основе анализа обсуждений в социальных сетях
9 Выявление аффилированности лиц на основе анализа данных СМИ
10 Оценка степени удовлетворенности клиентов услугами на основе обработки журнала событий контакт-центра
11 Обнаружение фактов инсайда по материалам социальных сетей и блогосферы
12 Выделение троллинга в интернет-данных
13 Поиск источников вброса, каналов и путей распространения информации в сети Интернет
14 Сбор и анализ данных по ценам и объёмам производства по основным позициям продовольственной безопасности на внешних рынках.
15 Выявление признаков недобросовестных организаций и методов их идентификации.
16 Разработка сервисов чтения и извлечения данных социальных сетей из распределённых файловых систем.
17 Анализ применимости технологий BigData для стратегического маркетинга новых товаров и услуг. (Агейкин М.А.)
18 Разработка лабораторной работы по сбору, анализу и визуализации данных с помощью IBM Data Explorer.
19 Определение целевой аудитории потребления товаров и услуг среди участников форумов и социальных сетей.
20 Создание команды специалистов - аналитиков и обработчиков данных (Data Scientists) под конкретную задачу с учетом существующей среды предприятия.
21 Методы управления командой аналитиков и обработчиков данных (Data Scientists) с учетом психологических и профессиональных профилей участников этой команды.
22 Разработка профилей руководителей и специалистов, в том числе команды Data Scientists, включая конкретные виды деятельности.
№ п/п
Тема междисциплинарной курсовой работы
23 Согласование профилей участников команды Data Scientists с профилем их руководителя
24 Применение методов экстремального управления (Agile) в работе команды Data Scientists.
25 Разработка методов геймификации в мобильном приложении анализа кардиограмм
26 Разработка методов геймификации в мобильном приложении оценки торговых точек
27 Разработка лабораторной работы по анализу и кластеризации данных и построению регрессионных моделей в IBM SPSS Statistics
28 Разработка лабораторной работы по сбору и обработке данных в IBM SPSS Modeler
29 Разработка лабораторной работы по построению моделей машинного обучения в IBM SPSS Modeler
30 Стандартизация адресов торговых точек: инструменты и методы
31 Исследование и создание технологии структуризация исходных данных на примере форм отчетности сельскохозяйственных предприятий
32 Построение витрины данных по формам отчетности сельскохозяйственных предприятий
33 Исследование и автоматизация процесса поиска рынков сбыта через открытые источники, в т.ч. форумы и социальные сети
34 Создание хранилища данных для анализа информации о торговых точках
35 Создание портала бизнес-аналитика для анализа потенциала торговых точек на base Cognos Analytics 11
36 Разработка лабораторных работ по задаче управления метаданными в среде хранилищ данных
37 Разработка лабораторных работ по noSQL базам данных
38 Разработка лабораторных работ по индексированию и поиску с использованием продуктов ElasticSearch
39 Разработка лабораторных работ по разработке портала для зада бизнес-анализа с использованием Cognos Analytics 11
40 Построение модели искусственной нейронной сети для распознавания ИБС по Фурье-спектру первого отведения ЭКГ.
41 Исследование задачи распознавания комплекс PQRST методами вейвлет-анализа
42 Исследование задачи регрессионного тестирования ETL-процессов
43 Исследование математических моделей зависимостей характеристик случайных величин применительно к результам нагрузочного эксперимента
44 Разработка лабораторной работы по определению целевой аудитории для товаров парфюмерной компании среди участников социальных сетей
47


ЕС-лизинг
Создание в ЗАО «ЕС-лизинг» и МИЭМ НИУ ВШЭ распределенного Центра компетенции по когнитивным технологиям IBM
IBM, ЗАО «ЕС-лизинг» и НИУ ВШЭ подписан Протокол о намерениях по созданию распределенного Центра компетенции по когнитивным технологиям с использованием облачных сервисов IBM Watson и облачной платформы IBM Bluemix
IBM Watson
IBM Bluemix
ЕС-лизинг
Стороны решили организовать сотрудничество в обучении студентов разработке и использованию облачных сервисов с привлечением ресурсов IBM, размещенных на университетском хабе «Академическая инициатива» и облачной платформе IBM Bluemix
НИУ ВШЭ стала одной из 535 организаций в мире, имеющей собственную страницу на Академическом хабе IBM в дополнение к Центру компетенции IBM Big Data (ЗАО «ЕС-лизинг» 2012г.)
48


ЕС-лизинг
2016: Новые методы работы с Большими Данными
НОВЫЕ МЕТОДЫ РАБОТЫ
С БОЛЬШИМИ ДАННЫМИ:
победные стратегии управления в бизнес-аналитике
Новые методы работы с большими данными: победные стратегии управления в бизнес-аналитике:
Научно-практический сборник.
Под редакцией доктора технических наук, профессора А.В.Шмида. –
М.: ПАЛЬМИР, 2016. – 528с.: илЛ.
Книга включает руководства по выполнению 10 лабораторных работ
Обобщен опыт 4-х лет обучения и написаны инструкций по подготовке магистров по технологии Big Data
49


ЕС-лизинг
Выводы
1. В ближайшие годы потребности в специалистах по Big Data по созданию ЭС составит не менее 80 - 100 тыс. человек в год (2020 – 2025гг. – 450 тыс. человек)
2. Необходимо готовить магистров во всех ВУЗах страны, обеспечив численность работающих в цифровой экономике
3. Комплексная подготовка магистров по этой специальности ведется на созданной в 2013 году базовой кафедре «Информационно-аналитические системы» в Национальном университете «Высшая школа экономики (МИЭМ)» (20 человек в год) на базе созданного в ЗАО «ЕС-лизинг» в 2012году совместно с компанией IBM Центра компетенции по продуктам IBM Big Data
IBM
The following attests that student
Kasimov Marat \ Kasimov Marat
has achieved an IBM Academic Qualification in course
"IBM BigData Platform"
IBM BigInsights, IBM iz, IBM Watson Content Analytics, IBM Watson Explorer
По завершению обучения на кафедре магистры получают Сертификат Академической программы IBM
50


ЕС-лизинг
С.Ю. Витте: «По головам или по стадам?» Так как же обеспечить численность работающих? Структура подготовки кадров цифровой экономики
Триада цифровой экономики
Сырье
Краулеры (добыча)
BIG Data
Завод
Обучаемые компьютеры
Товар
Решения
Логистика
Облачный доступ
Продажи
Мобильные приложения
По фирмам или по людям?
Кадры цифровой экономики
Учащиеся ПТУ
Бакалавры
Магистры
Аспиранты
Кандидаты наук, Доктора наук
Постановщики задач
срок подготовки - 4-5 лет
срок подготовки - 5 - 11 лет
массовая специальность срок подготовки - 3 года
Как?
Что?
Общее образование
Профильное образование
Оценка требований к численности работающих в цифровой экономике:
$30 трлн. / $1млн. (Google) = 30 млн. человек
5 % - 1.5 млн. человек
1,5% - 450 тыс. человек
Создать облачные Центры компетенции технологий Big Data для всех школ и ВУЗов страны
51


ЕС-лизинг
Главная стратегическая угроза!
«Хочешь победить врага? Воспитай его детей.»
Сунь-цзы. Трактат «Искусство войны»
Китай. V век до н.э.
Toy Story: IBM's Watson can now talk to your kids
Вместо родителей
http://venturebeat.com/2015/02/16/toy-story-ibms-watson-can-now-talk-to-your-kids/
52





В. И. Волчихин, А. М. Бершадский, А. С. Бождай

Пензенский государственный университет, Пенза, Россия

СЕРВИС-ОРИЕНТИРОВАННАЯ АРХИТЕКТУРА СИСТЕМЫ УПРАВЛЕНИЯ ЗНАНИЯМИ В ЭЛЕКТРОННЫХ ОБРАЗОВАТЕЛЬНЫХ СИСТЕМАХ[1]

Предлагается новый подход к построению всепроникающей единой сервис-ориентированной среды для оказания электронных образовательных услуг, обладающей такими свойствами, как территориальная инвариантность, круглосуточная доступность и самоадаптация к текущему уровню знаний обучаемого и доступных ему технических средств.

Анализ текущего состояния мирового рынка электронных образовательных систем позволяет сформулировать перечень основных требований к формам и методам дистанционного образования ближайшего будущего [1]:

[1] Работа выполнена при поддержке РФФИ (научный проект №15-07-01553).

- мобильность и оперативность образовательных технологий;
- адаптивность образовательных курсов и систем: как с точки зрения содержания, так и с точки зрения форм передачи знаний;
- сервис-ориентированность, позволяющая гибким образом подстраивать функционал систем дистанционного обучения (СДО) под требования каждого конкретного человека, а также эффективным образом использовать распределенные образовательные ресурсы.

Сегодня научное сообщество подошло к осознанию новой фазы электронного образования (E/U-Learning 3.0 – Electronic/Ubiquitous Learning – всепроникающее электронное обучение)), в которой важную роль будут играть распределенные компьютерные системы, облачные технологии, мобильные персональные устройства, системы искусственного интеллекта и средства виртуальной реальности. Другой важной проблемой является многоуровневая самоорганизация (адаптируемость) компонентов электронной образовательной среды без лишних процессов перекомпиляции, что приведет к их универсализации и продлению жизненного цикла.

Таким образом, создание СДО, обладающих свойствами мобильности, адаптивности и сервис-ориентированности является в настоящее время актуальной научно-практической проблемой и перспективным направлением развития электронного образования в целом. Схема связей между целями и задачами систем E/U-Learning с одной стороны и технологическими методами их достижения – с другой, показаны на рис. 1.



Рис. 1. Основные технологические составляющие методов управления знаниями в системах E/U-Learning

Ключевым технологически аспектом создания и использования всепроникающих СДО является их самоадапируемость к целому ряду внешних и внутренних условий. Свойство самоадаптируемости является определяющим при решении задач расширения жизненного цикла программных компонентов СДО. Поскольку всепроникающие технологии обучения подразумевают непрерывный цикл оказания дистанционных образовательных услуг, то самоадаптация должна быть прозрачной для конечного пользователя и проходить в реальном режиме времени без остановки функционирования на время перекомпиляции и отладки новых версий.

К настоящему времени, задача самоадаптации может быть решена с применением технологии DSPL (Dynamic Software Product Lines – динамические линейки программных продуктов) [2]. В этом случае программный продукт может рассматриваться не в контексте одной лишь текущей версии, жизненный цикл которой заканчивается с появлением новой версии, а в контексте совокупности всех возможных модификаций и изменений продукта в зависимости от условий окружающей среды и требований пользователя. Неотъемлемой частью такого продукта должна быть модель изменчивости предметной области, заранее предусматривающая и формализующая различные допустимые варианты ее изменения. Далее, на основе возникновения новых вариантов предметной области, генерируется новая конфигурация программной части продукта.

В данной работе предлагается концептуальная структура всепроникающей СДО, включающую как традиционные компоненты электронного образования, так и компоненты, обеспечивающие управление изменчивостью и самоадаптацией (рис. 2).

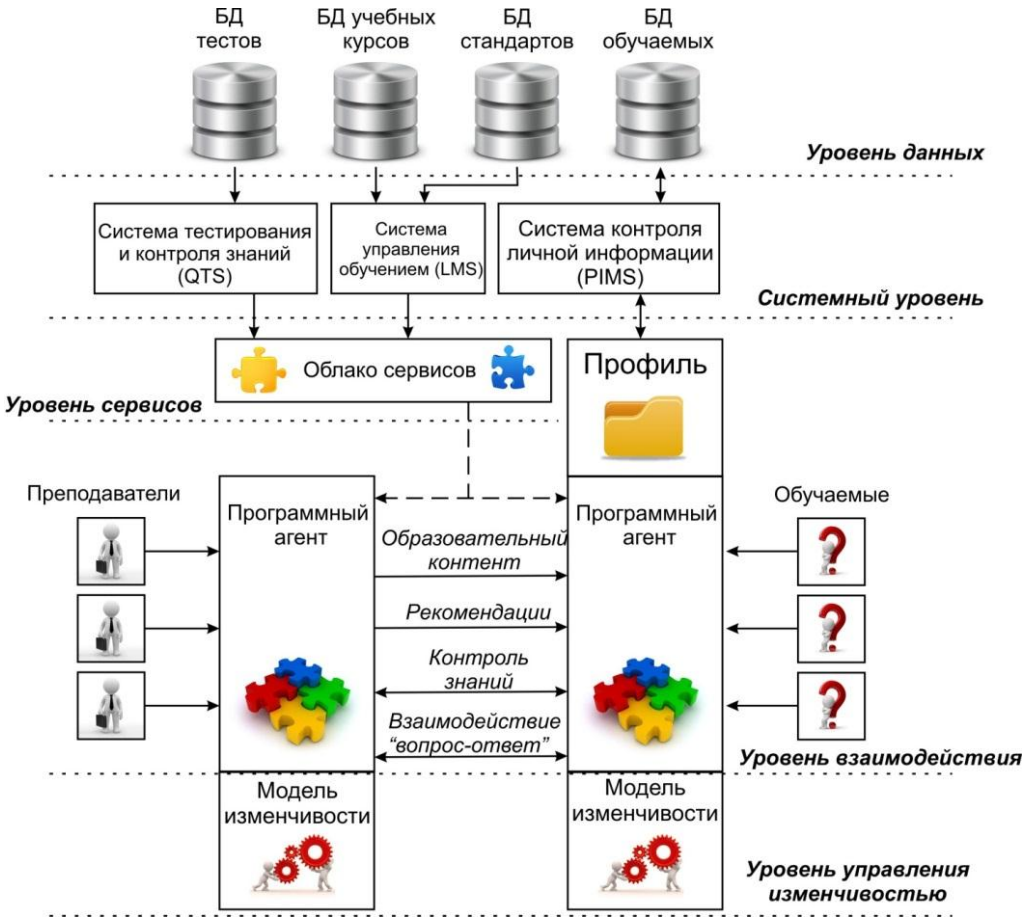


Рис. 2. Сервис-ориентированная архитектура системы управления потоками знаний

В предлагаемой структуре можно выделить 5 основных уровней:

– Уровень взаимодействия (Level of Interaction). Обеспечивает взаимодействие основных участников образовательного процесса: студента и преподавателя посредством интеллектуальных программных агентов. Программные агенты являются интерфейсной, т.е. видимой для участника образовательного процесса, частью системы E/U Learning. Агент обучаемого это интеллектуальная программная оболочка, устанавливающая формат и выборку образовательного контента, определяющая интерфейс для его визуализации, выполняющая адаптацию контента для различных видов программно-аппаратных платформ и каналов передачи данных. Источником контента для агента обучаемого является потоки знаний, получаемых от агента преподавателя – интеллектуальной программной оболочки,

предоставляющей преподавателю доступ к функциям систем управления обучением (LMS) и тестирования (QTS). Программные агенты являются самоадаптирующейся частью системы E/U Learning. Их конфигурация собирается из набора отдельных web-сервисов под управлением соответствующих моделей изменчивости (VM).

– Уровень управления изменчивостью (Level of Variability). Включает модели изменчивости, управляющие текущими конфигурациями программных агентов.

– Уровень сервисов (Level of Services). Включает базовый набор web-сервисов, из которых выстраиваются конфигурации программных агентов и предоставляется доступ к системам функционального уровня.

– Функциональный уровень (Level of Systems). Включает основные системы управления образовательным процессом, в том числе: систему управления обучением (LMS), систему тестирования и контроля знаний (QTS), систему контроля личной информации (PIMS). LMS отвечает за управление контентной составляющей программных агентов в рамках отдельных учебных курсов. QTS обеспечивает процессы промежуточного тестирования и передачу их результатов сначала в LMS и PIMS, а затем оттуда осуществляется связь с уровнем управления изменчивостью для изменения конфигурации программного агента обучаемого.

– Уровень данных (Level of Data). Включает базы данных для информационного обеспечения систем функционального уровня. База модулей курсов и база образовательных стандартов обеспечивает LMS. База тестов обеспечивает QTS. База студентов с личной информацией, траекториями обучения и текущей успеваемостью связана с PIMS.

На рис. 3 показан процесс взаимодействия модели изменчивости и конфигураций программного агента обучаемого в зависимости от успешности прохождения учебных курсов (по результатам тестирования и индивидуальным ответам на вопросы преподавателя), доступных технических и телекоммуникационных возможностей, психофизических особенностей студента. На основе анализа текущего профиля студента, выбирается оптимальная конфигурация программного агента, определяемая технической, интерфейсной и контентной составляющими.

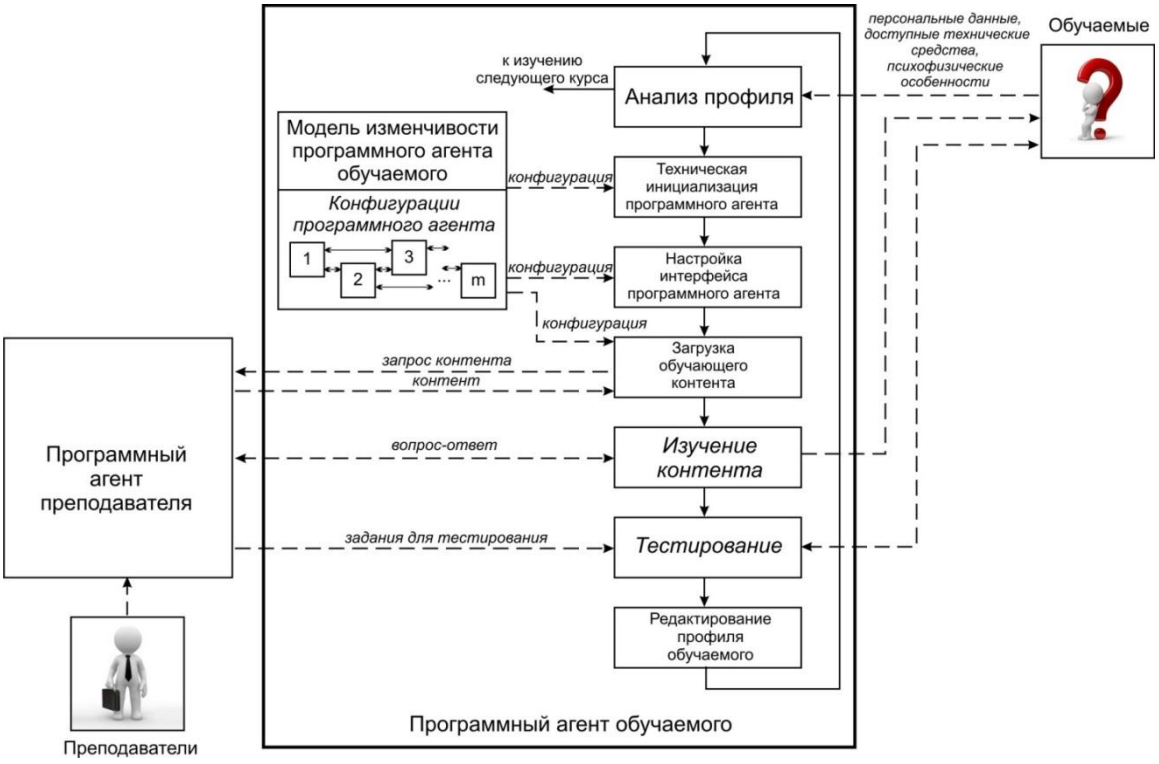


Рис. 3. Процесс взаимодействия модели изменчивости и конфигураций программного агента обучаемого

Предложенный в статье подход позволяет подойти к решению фундаментальной проблемы создания единой межгосударственной интеллектуальной среды для оказания электронных образовательных услуг, в которую будут включены гетерогенные формы представления знаний, межгосударственные стандарты и формы обучения, международный преподавательский и студенческий состав. Построение такой среды открывает новую фазу развития электронного образования (E/U-Learning 3.0), в которой ведущую роль будут играть распределенные компьютерные системы, облачные технологии, мобильные персональные устройства, системы искусственного интеллекта и средства виртуальной реальности.

Библиографический список

1. Бершадский, А. М. Принципы построения общедоступной самоадаптирующейся системы дистанционного обучения на основе модели изменчивости и сервис-ориентированной архитектуры / А. М. Бершадский, А. С. Бождай, В. С. Мкртчян // Информационные технологии. – 2016. – № 2. – С. 154–160.

2. Papazoglou, M. P. Service oriented architectures: approaches, technologies and research issues / M. P. Papazoglou, W.-J. Heuvel // The VLDB Journal. – 2007. – Vol. 16, № 3. – P. 389–415.

Т. В. Глотова, И. Г. Кревский

Пензенский государственный университет, Пенза, Россия

ИСПОЛЬЗОВАНИЕ ВОЗМОЖНОСТЕЙ ЭЛЕКТРОННОЙ ИНФОРМАЦИОННО-ОБРАЗОВАТЕЛЬНОЙ СРЕДЫ УНИВЕРСИТЕТА В ПРЕПОДАВАНИИ IT ДИСЦИПЛИН

Функционирование электронной информационной образовательной среды и применение современных дистанционных образовательных технологий предоставляют дополнительные возможности для эффективной организации образовательного процесса, использования активных и интерактивных методов обучения, реализации индивидуального подхода к обучению.

Применение современных дистанционных технологий в системе высшего образования приводит к трансформации традиционного обучения в смешанное (blending learning), которое комбинирует традиционное обучение и дистанционные образовательные технологии. Даже в нормативных документах термин «аудиторные занятия» с апреля 2017 г. заменен на «контактная работа», которая «может быть аудиторной, внеаудиторной, а также проводиться в электронной информационно-образовательной среде» [1]. Таким образом, контактная работа включает обучение как «лицом к лицу» с преподавателем в аудитории, так и дистанционно в виртуальном образовательном пространстве. Переход на ФГОС требует для реализации образовательных программ функционирования в университете электронной информационной образовательной среды, в которой должны быть развернуты все учебные и методические ресурсы, необходимые для подготовки студентов, обеспечен доступ ко всей литературе, указанной в рабочих программах дисциплин, фиксация хода образовательного процесса, взаимодействие преподавателей и студентов. Таким образом, при выполнении требований ФГОС ВО компоненты электронного обучения и дистанционных технологий должны быть внедрены в любую образовательную программу независимо от

формы обучения [2]. В настоящее время в Пензенском государственном университете идет внедрение ЭИОС в образовательный процесс. ЭИОС реализуется с использованием системы управления учебным процессом (Learning Management System – LMS) Moodle, ранее используемой только в дистанционном обучении. Поэтому актуален вопрос о внедрении возможностей ЭИОС в образовательный процесс университета, прежде всего использовании возможностей Moodle в методике организации занятий со студентами.

При традиционной организации учебного процесса на сегодняшний день имеются следующие проблемы [3]:

- уменьшение в учебных планах количества аудиторных занятий по дисциплинам;
- необходимость увеличения интенсивности подачи материала на занятиях в связи с увеличением объема информации в программах дисциплин информационной направленности, так как образовательный процесс идет в условиях быстро меняющейся экономической ситуации и стремительного развития техники;
- неумение большей части студентов работать самостоятельно без– постоянного контроля преподавателя, особенно в тех видах деятельности, где требуются уровни не просто понимания, а трансформации знаний и систематической творческой работы;
- необходимость постоянного мотивирования обучающихся.

Для преодоления этих проблем можно использовать возможности систем управления обучением.

Для реализации дисциплины с использованием возможностей LMS Moodle необходимо размещение в раздел дисциплины в Moodle рабочей программы дисциплины, лекционных материалов и практических (лабораторных) заданий, контрольно-тестовых материалов (при наличии). Раздел дисциплины в Moodle фактически является электронным вариантом учебно-методического комплекса (УМК) и включает методические рекомендации по курсу, разъясняющие вопросы порядка возможного изучения тем, описание взаимосвязи теоретических и практических материалов, контрольные точки, требования для прохождения текущих и промежуточной аттестаций, способы взаимодействия с преподавателем, задания для групповой работы (если она предусмотрена), графики традиционных и онлайн консультаций, например в чате. Студенту необходимо обеспечить доступ ко всей основной и дополнительной литературе, рекомендованной в рабочей программе, соответственно вся литература должна либо иметься в больших количествах в вузовской библиотеке, либо должна быть доступна по ссылкам в разделе дисциплины. При этом физически электронные издания могут размещаться в электронной библиотеке вуза, быть открытыми образовательными ресурсами, размещаться в электронных библиотеках, с которыми у вуза есть договор об их использовании.

Возможность предварительного размещения учебного материала и ссылок на литературу в разделе LMS Moodle дает возможность применения активных и интерактивных методов обучения, например, организации лекций в форме обсуждений и дискуссий. В течение семестра возможно добавление материалов в ходе дискуссии или при появлении новых материалов по теме дисциплины. В некоторых вузах [4] это позволяет осуществить перевод изучения отдельных дисциплин в формат blended learning, когда лекционная часть изучается дистанционно, а семинары посвящены обсуждениям и практической отработке изученного онлайн материала.

Возможности LMS Moodle для проведения тестирования обеспечивают оперативность текущего контроля и автоматизации обработки результатов. Многообразие типов вопросов, предоставляемые системой, дает возможность составления надежных и валидных тестов. Но необходимо отметить, что разработка тестов – это трудоемкий и долговременный процесс, требуемый исследования результатов тестирования реальных групп студентов. Известные ограничения применения тестирования для проверки практических навыков и умений, а также решения задач требуют разработки специализированных интерактивных тренажеров и виртуальных лабораторий в соответствующих областях [5–6]. Для проверки

заданий, которые не поддаются автоматической оценке, можно использовать возможность оценки и рецензирования работы самими студентами, что дает студентам опыт анализа и более высокий уровень усвоения материала. Конечно, компьютерное тестирование в ближайшем будущем не заменит опытного и квалифицированного преподавателя, использующего технологии обучения через вопросы и задачи, проблемный и проектный подход, но обеспечит широкий охват студентов и оперативность контроля знаний. Особенно это важно для контроля самостоятельной работы студентов, объем которой в современных программах значительно увеличился.

Большие возможности интерактивного взаимодействия студентов и преподавателей предоставляют интерактивные форумы, которые используются для организации онлайн и офлайн общения студентов и преподавателей. Для интерактивного обсуждения и оперативного выяснения вопросов курсового проектирования рекомендуется создать форум в разделе дисциплины в ЭИОС университета. Преподаватель может в форуме разместить ответы на часто задаваемые вопросы по выполнению курсового проекта или лабораторных работ. На первых этапах, возможно, потребуется дополнительная мотивация студентов для участия в интерактивных обсуждениях, которая может быть организована за счет выделения специальных баллов в балльно-рейтинговой системе дисциплины. Необходимо отметить, что учет работы студентов в Moodle ведется автоматически.

Для реализации индивидуального подхода к сильным студентам, для магистрантов и аспирантов в разделе дисциплины имеется возможность разместить ссылки на учебные материалы лучших университетов мира, массовые, открытые для всех онлайн курсы (МООК), социальные сети программистов и профессионалов других направлений, специально созданные для обсуждения профессиональных вопросов.

Наличие ЭИОС университета предоставляет широкий диапазон возможностей использования информационно-коммуникационных технологий в учебном процессе и их грамотное сочетание с педагогическими технологиями образовательного проектирования позволит обеспечить качество обучения в вузе на современном уровне.

Библиографический список

1. Об утверждении Порядка организации и осуществления образовательной деятельности по образовательным программам высшего образования – программам бакалавриата, программам специалитета, программам магистратуры : приказ Минобрнауки России № 301 от 05.04.2017.

2. Бершадский, А. М. Выполнение требований ФГОС 3+ – шаг в развитии электронного обучения / А. М. Бершадский, Т. В. Глотова, И. Г. Кревский // Информационное общество: образование, наука, культура и технологии будущего : сб. тр. XVIII объединенной конф. (IMS-2015) (Санкт-Петербург, 23–25 июня 2015 г.). – СПб. : Университет ИТМО, 2015. – С. 21–32.

3. Бершадский, А. М. Современный университет: образовательные и информационные технологии в многоуровневой системе высшего образования / А. М. Бершадский, Т. В. Глотова, И. Г. Кревский // Информационное общество: образование, наука, культура и технологии будущего : сб. науч. ст. XVIII объединенной конф. (IMS-2016), (Санкт-Петербург, 21–24 июня 2016 г.). – СПб. : Университет ИТМО, 2016. – С. 143–160.

4. Кулик, Е. Ю. Вышка онлайн: стратегии и перспективы образовательных технологий / Е. Ю. Кулик // Окна академического роста. – URL:https://okna.hse.ru/news/162195755.html

5. Лисицына, Л. С. Методы и алгоритмы теории графов Открытое образование : онлайн-курс / Л. С. Лисицына. – URL: https://openedu.ru/course/ITMOUniversity/AGRAPH/ (дата обращения: 07.03.2016).

6. Якименко, О. В. Применение обучающих программ-тренажеров в обучении программированию / О. В. Якименко, А. Н. Стась // Вестник Томского государственного педагогического университета. – 2009. – № 1. – URL: http://cyberleninka.ru/article/n/ primenenie-obuchayuschih-programm-trenazherov-v-obuchenii-programmirovaniyu (дата обращения: 22.04.2016).

З. И. Баусова, О. В. Прокофьев, Э. Ф. Шадрина

Пензенский государственный университет, Пенза, Россия

АНАЛИЗ ПРОГРАММНЫХ ПРОДУКТОВ ЭКОНОМЕТРИЧЕСКОГО МОДЕЛИРОВАНИЯ В УЧЕБНОМ ПРОЦЕССЕ

Содержатся результаты анализа бесплатных программных продуктов для эконометрического моделирования. В работе представлены данные о ценовых и функциональных возможностях различных программных продуктов.

В настоящее время в России большинство программных продуктов, применяемых в образовательных целях и проведении научных исследований, для образовательных учреждений, в том числе и высших учебных заведений (вузов) предоставляются со скидками (по инициативе фирм-производителей) или бесплатно. На данный момент задача развития программы поддержки вузов очень актуальна, поскольку внедрение программных продуктов для статистического и эконометрического моделирования является довольно затратным мероприятием [3, 4].

Таким образом, для одного компьютера лицензия на программный пакет IBM SPSS Statistics Standard (позволяет использовать основной аналитический функционал для широкого применения в бизнесе и исследования), предоставляемая на 12 месяцев, стоит от 202 752 руб. и, в зависимости от версии, выше. Например, программный пакет Statistica Enterprise 13 English можно приобрести для коммерческих организаций и вузов (1 пользователь, бессрочно) по цене с экномией 4 % от 1 675 934 руб., когда цена для прочих организаций составляет 1 739 177 руб. Но есть и заманчивые предложения, такие как версия Statistica Ultimate 13 English (локальная лицензия для образовательного учреждения – 1 пользователь) Statistica ULTIMATE ACADEMIC BUNDLE 13 English for Windows на одного пользователя, предоставляемая по цене также с экономией 4 % (реальная цена 9392 руб.) 9 050.00 руб. (цена для образовательных учреждений). Существуют также пробные trial-версии программных продуктов, которые можно найти в свободном доступе в сети Интернет на сайте компаний-производителей. Срок их действия составляет 14 дней, что очень мало для процесса обучения, соответственно, они не пригодны для учебного процесса. Вся суть вопроса состоит в том, что лицензии для одного рабочего места, что очевидно, для обеспечения полноценного учебного процесса недостаточно. Как правило, образовательному учреждению, в среднем, необходимо купить от 20 до 80 лицензий, в зависимости от масштабов учреждения. Это очень затратно и зачастую не по силам образовательным учреждениям из-за большого количества платного лицензионного программного обеспечения, требуемого в процессе обучения. Например, стандартные программные пакеты Microsoft Office Word, Microsoft Office Excel, Microsoft Office Power Point и многие другие. Образовательным учреждениям не приходится ждать поддержки от государства, поэтому возникает потребность искать альтернативные программные пакеты, не требующие покупки лицензий. Таким образом, проблема платности программных продуктов является актуальной проблемой современности.

Существует также универсальная общедоступная лицензия GNU (General Public License) – это лицензия на свободное программное обеспечение, созданная в рамках проекта GNU в 1988 г., по которой автор передает программное обеспечение в общественную собственность. Ее цель состоит в том, чтобы предоставить пользователю права копировать, модифицировать и распространять (в том числе на коммерческой основе) программы (например, программы с открытым кодом), а также гарантировать, что и пользователи всех

производных программ получат вышеперечисленные права. Данная лицензия не позволяет делать с программами все что угодно. В основе данной лицензии заложен принцип «наследования» прав называется «копилефт» и был придуман Ричардом Столлмэном. По контрасту с GPL, лицензии собственнического программного обеспечения очень редко дают пользователю такие права и обычно, наоборот, стремятся их ограничить, например, запрещая восстановление исходного кода. Лицензируя работу на условиях GNU GPL, автор сохраняет за собой авторство [10].

Среди пакетов для эконометрического моделирования существуют пакеты, обладающие открытым, свободным и бесплатным программным обеспечением со стандартной общественной лицензией GNU, при этом за качество и точность функционирования пакета отвечает пользователь. Ими являются продукты R и GretL. Рассмотрим бесплатно и свободно распространяемые программные пакеты:

R – статистическая система анализа, созданная Россом Ихакой и Робертом Гентлеманом (1996, J.Comput. Граф. Stat., 5: 299-314). R является и языком и программным обеспечением; его наиболее замечательные особенности:

• эффективная обработка данных и простые средства для сохранения результатов;

• набор операторов для обработки массивов, матриц, и других сложных конструкций;

• большая, последовательная, интегрированная коллекция инструментальных средств для проведения статистического анализа;

• многочисленные графические средства.

R – язык, позволяющий пользователю использовать операторы циклов, чтобы последовательно анализировать несколько наборов данных. Также возможно объединить в отдельную программу различные статистические функции, для проведения более сложного анализа [5, 6].

GretL – это кросс-платформенный программный пакет для эконометрического анализа, написанный на языке Си. GretL является открытым, свободным и бесплатным программным обеспечением. Можно распространять и/или модифицировать его согласно лицензии GNU General Public License (GPL), текст которой предоставлен Free Software Foundation [1, 2, 7].

Особенности GretL:

- простой и интуитивно понятный интерфейс на всех языках;
- множество методов оценивания, инструментарий для анализа временных рядов;
- использование моделей с ограниченной зависимой переменной;
- выдача результатов в формате LaTeX в форме таблицы или уравнения одним нажатием кнопки;
- мощный встроенный скриптовый язык hansl, содержащий большое количество функций для программирования и работы с матрицами;
- интеграция и обмен данными со статистическими пакетами GNU R, GNU Octave, Ox и Stata.

PAST 3.14 – простой пакет в использовании инструмента анализа данных, который позволяет обрабатывать статистические данные, создавать графики и рассчитать различные статистические показатели [9].

Программа имеет портативную версию и не требует инсталляции. На сайте компании-производители имеется полезное руководство в формате *.pdf и дополнительные обучающие материалы и примеры. Периодически авторы совершенствуют программу PAST 3.14 и расширяют ее возможности. Значительным минусом данного программного обеспечения является его англоязычность и неадаптированность к интернациональным пользователям.

Универсальный российский статистический пакет STADIA – за 12 лет существования и развития стал аналитическим инструментом для многих тысяч пользователей в раз-

личных областях науки, техники, планирования, управления, производства, сельского хозяйства, экономики, бизнеса, маркетинга, образования, медицины и др. по всей русскоязычной Евразии. По своим базовым возможностям сопоставим с наиболее известными западными статистическими пакетами. Отличается познаваемостью и простотой использования применительно к отечественной аудитории [8].

В заключении можно сделать вывод о том, что наиболее широкими и безграничными функциональными возможностями из бесплатных программных средств обладает среда R, имеющая 12600 пакетов расширений. Однако, для использования программного пакета R необходимы значительные навыки программирования, что доступно не всему контингенту желающих моделировать в данной программной среде. Приемлемыми в применении являются такие программные пакеты, как GretL (частично требующий знаний программирования) и полностью русифицированный пакет для эконометрического моделирования Stadia. Программный пакет PAST 3.14 неплох в функциональном плане, но, к сожалению, совсем не предусмотрен для пользователей, знающих исключительно только русский язык.

Библиографический список

1. Шадрина, Э. Ф. К вопросу выбора прикладных программ для решения эконометрических задач в области интеллектуальной собственности / Э. Ф. Шадрина, О. В. Прокофьев // Информационные технологии в науке и образовании. Проблемы и перспективы : сб. науч. ст. IV ежегодной межвуз. науч.-практ. конф. – Пенза : Изд-во ПГУ, 2017. – С. 226–228.

2. Баусова, З. И. Оценка эффективности экономической информационной системы (эис) обработки агентских платежей за энергопотребление / З. И. Баусова, Г. В. Бобрышева, А. Ю. Старикова, Э. Ф. Шадрина // Информационные технологии в науке и образовании. Проблемы и перспективы : сб. науч. ст. IV ежегодной межвуз. науч.-практ. конф. – Пенза : Изд-во ПГУ, 2017.

3. Баусова, З. И. Применение эконометрического моделирования к анализу патентных изобретении / З. И. Баусова, О. В. Прокофьев, А. Ю. Старикова, Э. Ф. Шадрина // Новые информационные технологии и системы : сб. науч. ст. XIII Междунар. науч.-техн. конф. – Пенза : Изд-во ПГУ, 2016. – С. 51–53.

4. Старикова, А. Ю. К вопросу применения имитационного моделирования для управления интеллектуальной собственностью / А. Ю. Старикова, З. И. Баусова, Э. Ф. Шадрина // Новые информационные технологии и системы : сб. науч. ст. XII Междунар. науч.-техн. конф. – Пенза : Изд-во ПГУ, 2016. – С. 264–265.

5. Баусова, З. И. Эконометрическое моделирование в среде R для учебной и научно-исследовательской работы в магистратуре / З. И. Баусова, А. Ю. Старикова, О. В. Прокофьев // Информационные технологии в науке и образовании. Проблемы и перспективы : сб. науч. ст. IV ежегодной межвуз. науч.-практ. конф. – Пенза : Изд-во ПГУ, 2017. – С. 117–119.

6. Информационный ресурс компании-производителя программной среды R. – URL: http://r-analytics.blogspot.ru/p/blog-page_20.html#.WOy6UtSLTGg

7. Информационный ресурс компании-производителя программной среды GretL. – URL: http://gretl.sourceforge.net/ru.html#man

8. Информационный ресурс компании-производителя программной среды Stadia. – URL: http://old.exponenta.ru/soft/others/stadia/stadia.asp

9. Информационный ресурс компании-производителя программной среды Pest. – URL: http://www.filecluster.com/Home-Education/Science/Download-PAST.html

10. Википедия – свободная энциклопедия. – URL: https://ru.wikipedia.org/wiki/GNU_G

11. Баусова, З. И. К вопросу о применении имитационного моделирования в выпускной квалификационной работе бакалавров (по направлению «прикладная информатика») / З. И. Баусова, Г. В. Бобрышева, А. Ю. Старикова, О. В. Прокофьев // Университетское образование (МКУО–2016) : сб. ст. XX Междунар. науч.-метод. конф. – Пенза : Изд-во ПГУ, 2016. – С. 84–85.

И. А. Ермолов

Ростовский государственный экономический университет,
Ростов-на-Дону, Россия

РАЗРАБОТКА WEB-ОРИЕНТИРОВАННОЙ ЭКСПЕРТНОЙ СИСТЕМЫ ПОДДЕРЖКИ ТРУДОУСТРОЙСТВА ВЫПУСКНИКОВ ВУЗА С ПРИМЕНЕНИЕМ ТЕХНОЛОГИЙ SEMANTIC WEB

Представлена модель архитектуры информационной системы поддержки трудоустройства выпускников вуза, разрабатываемой на основе онтологического инжиниринга и технологий Semantic Web. База знаний системы представлена в виде RDF-графа, к которой применяются Sparql-запросы. Система предоставляет возможность гибкой модификации и расширения базы знаний новыми данными, которые связываются между собой в динамическом режиме.

Одной из важнейшей задачей вузов является трудоустройство выпускников. В связи с этим актуальной является задача создания web-ориентированной информационной системы для поддержки эффективного взаимодействия вуза с его выпускниками и потенциальными работодателями [1].

Создаваемая система позволяет выпускнику трудоустроиться по специальности, а вузу оценить качество реализуемых образовательных программ и востребованность образовательных компетенций. Работодатель, в свою очередь, получает возможность удаленного поиска кандидатов среди выпускников, а также оценку их знаний и умений. Также система имеет возможность предоставления данных других внешним системам, с помощью веб сервисов [2]. Как работодателям, так и выпускникам будет доступна процедура автоматического подбора подходящих вакансий/резюме.

Центральным звеном создаваемой системы является сервис подбора работодателями наиболее подходящих кандидатов среди выпускников вуза. Данная компонента представляет собой встроенную экспертную систему, которая на основе диалога с работодателем в автоматическом режиме позволяет связать квалификационные требования к выпускнику с освоенными выпускником компетенциями, его знаниями, умениями и навыками. Система автоматически подберет по заданным работодателем требованиям наиболее подходящие кандидатуры выпускников с нужными компетенциями, соответствующие запросам пользователя.

Для разработки системы экспертной системы используются технологии Semantic Web. Средствами онтологического моделирования создается структурированное хранилище информации в формате RDF и множество правил вывода, которые информационный сервис может использовать для эффективной организации и поиска релевантных данных [3]. В результате такого описания предметная область взаимоотношений работодателей с выпускниками представляется как иерархическая структура, состоящая из множества информационных объектов, с которыми можно выполнять такие операции, как семантический поиск, агрегация данных, вывод и формулировка новых знаний [4].

В онтологической модели мы рассматриваем в виде классов множество образовательных программ и профессиональных стандартов, и составляющих их образовательных (профессиональных) компетенций, а также отношения между ними (в том числе отношений использования, соответствия). Отношения между классами помогут далее при реализации информационных запросов пользователей определить степень соответствия каждой образовательной программы профессиональному стандарту, а каждого выпускника конкретным требованиям работодателя. А также вместе с правилами логического вывода используются для реализации автоматического подбора вакансий/резюме.

Для каждой образовательной программы в онтологической модели приводится набор образовательных компетенций. Каждая компетенция из образовательной программы связывается с набором знаний, умений, формируемых дисциплинами учебного плана. Знаниям и умениям из образовательной программы, и требованиям профессионального стандарта назначается словарь синонимов, позволяющих осуществлять информационный поиск на «языке» работодателя.

Для создания базы знаний информационной системы поддержки процессов взаимодействия выпускников и работодателей нами был использован редактор онтологий Protégé 5.0. Построенная на основе структуры образовательных специальностей и профессиональных стандартов онтология определяет общий тезаурус для вуза и работодателей, совместно использующих информацию о компетенциях выпускников в данной предметной области. На рис. 1 приведен фрагмент онтологической модели ИС, в виде компетенций образовательных программ вуза, соотнесенных с требованиями профессиональных стандартов в сфере IT.

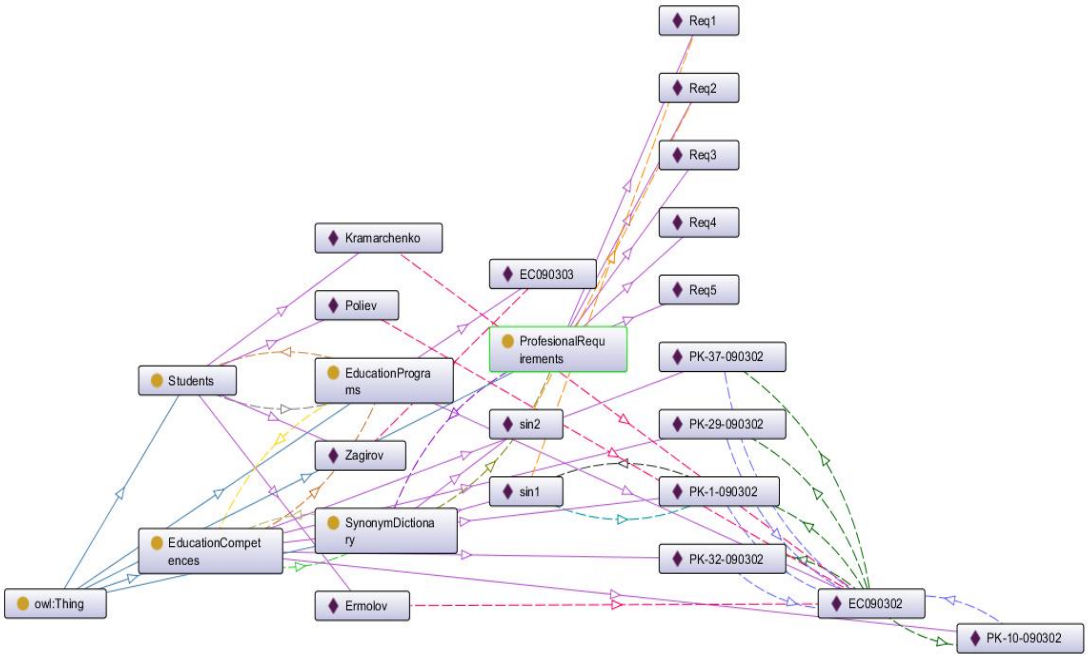


Рис. 1. Фрагмент онтологической модели ИС

Созданная модель описывают структуру компетенций образовательных программ в соответствии с требованиями к специалисту, заложенными в профессиональных стандартах. В качестве хранилища данных используется Jena Fuseki Web Server, в котором хранится база знаний в виде RDF утверждений. RDF – это данные, утверждения, представленные в виде триплетов «субъект-предикат-объект» [5]. Для идентификации субъектов и предикатов используется идентификатор Internationalized Resource Identifier (IRI), являющийся обобщением понятия URL.

Для разработки информационной системы используется язык программирования JAVA, в частности Jena Framework. Он предоставляет возможность работать непосредственно с RDF-хранилищем через язык запросов SPARQL.

SPARQL – язык запросов к RDF-хранилищам, который был принят в качестве стандарта W3C. Запрос на SPARQL представляет собой шаблон, описывающий подграф, который требуется найти или изменить в общем графе онтологии, представленной в RDF-формате.

На рис. 2 приведен Sparql-запрос, построенный по запросу, введенному в системе работодателем, на поиск студентов, которые владеют профессиональными компетенциями, необходимыми для выполнения функции «Инженерно-техническая поддержка подготовки

коммерческого предложения заказчику на создание (модификацию) и ввод в эксплуатацию типовой ИС на этапе предконтрактных работ».

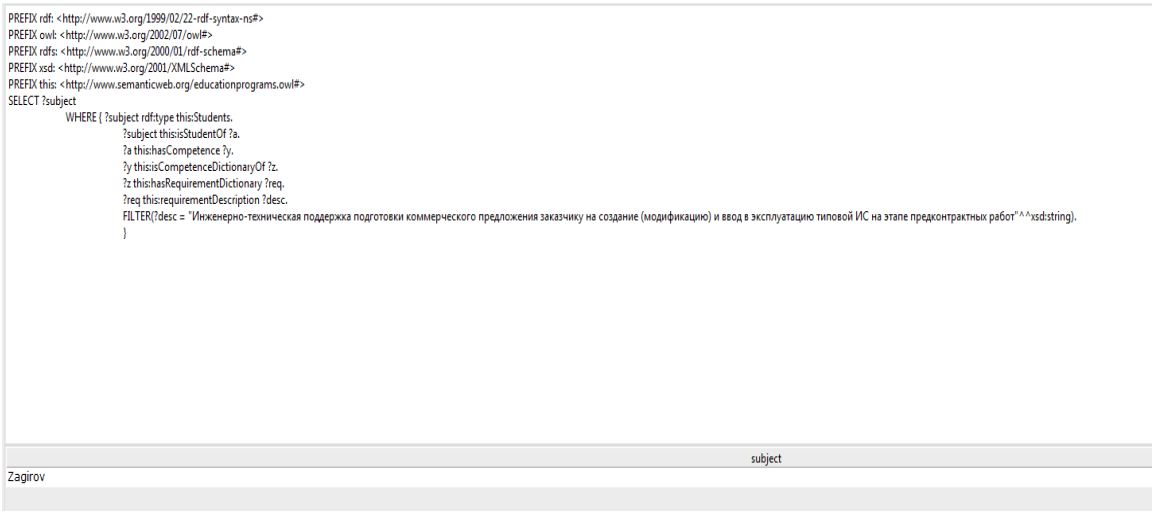


Рис. 2. Sparql-запрос, построенный по запросу работодателя на поиск студентов с требуемыми компетенциями

Работа экспертной системы по автоматическому подбору кандидатов представлена в виде следующей схемы:

- загрузка запроса в систему через интерфейс работодателя;
- преобразование запроса в Sparql-запрос;
- поиск соответствия в онтологии Профстандартов;
- поиск соответствия в онтологии Образовательных программ;
- поиск соответствия в онтологии Выпускников.

Архитектура разработанной ИС является расширяемой с возможностью замены или добавления отдельных компонентов системы. Программные модули являются независимыми от данных, что позволяет системе работать с новыми данными, дополненными из других предметных областей, по новым специальностям и образовательным программам. Объединение онтологий различных специальностей создает единую информационно-образовательную среду вуза [6].

В результате процесс разработки становится более гибким, а сама ИС более адаптивной к изменяющимся требованиям работодателей и условиям рынка труда.

Библиографический список

1. Шполянская, И. Ю. Анализ требований работодателей в системе поддержки трудоустройства выпускников вуза / И. Ю. Шполянская, Н. Ю. Мисиченко // Вестник Ростовского государственного экономического университета (РИНХ). – 2009. – № 28. – С. 330–337.

2. Шполянская, И. Ю. Референтная онтологическая модель бизнеса как основа создания WEB-ориентированных систем и сервисов / И. Ю. Шполянская // Вестник Ростовского государственного экономического университета (РИНХ). – 2015. – № 2 (50). – С. 220–226.

3. Загорулько, Ю. А. Построение порталов научных знаний на основе онтологий / Ю. А. Загорулько // Вычислительные технологии. – 2007. – С.П. 2. – Т. 12.

4. Добров, Б. В. Онтологии и тезаурусы: модели, инструменты, приложения : учеб. пособие / Б. В. Добров, В. В. Иванов, Н. В. Лукашевич, В. Д. Соловьев. – М. : Интернет-Университет Информационных технологий; Бином. Лаборатория знаний, 2009. – 173 с.

5. Resource Description Framework (RDF): концепции и абстрактный синтаксис / под ред. Дж. Клайн, Дж. Дж. Кэрролл ; Рекомендация W3C. – URL http://www.w3.org/TR/rdf-concepts/

6. Шполянская, И. Ю. Онтологическая модель информационно-образовательной среды учебного подразделения вуза / И. Ю. Шполянская, И. И. Мирошниченко // Вестник Ростовского государственного экономического университета (РИНХ). – 2010. – № 31. – С. 201–209.

Д. М. Жук, С. Ю. Князева, В. Б. Маничев, Д. И. Оглоблин

Московский государственный технический университет
им. Н. Э. Баумана, Москва, Россия

ОБУЧЕНИЕ БАКАЛАВРОВ ОСНОВАМ СИСТЕМ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ В ТЕХНИЧЕСКОМ ВУЗЕ

Обсуждается методика обучения бакалавров на основе CAD/CAM/CAE-технологий на втором курсе технического вуза.

Введение

В условиях рыночной экономики актуальность темы обучения бакалавров системам автоматизированного проектирования (САПР) приобретает все большее значение [1, 2]. Учебные программы подготовки бакалавров инженерных дисциплин предусматривают преподавание общеобразовательных предметов на младших курсах и профилирующих на старших курсах. Фактическое сокращение сроков обучения привело к уменьшению часов на изучение САПР как отдельной дисциплины. Тем не менее, предприятия, ведущие разработки без использования САПР, оказываются неконкурентоспособными, а выпускники вузов, не умеющие работать в современных САПР и не владеющие соответствующими знаниями, занимают низшие позиции на рынке труда.

Текущее состояние обучения САПР

Бакалавры на первом курсе изучают основы черчения и инженерной графики на бумаге и с привлечением САПР геометрического моделирования, которые относятся к группе 2D/3D CAD систем. Начальной средой обучения обычно является, либо AutoCAD-система автоматизированного 2D/3D проектирования и черчения, разработанная компанией Autodesk, либо Autodesk Inventor – система автоматизированного 3D твердотельного и поверхностного параметрического проектирования той же компании. Студенты выполняют типовые машиностроительные чертежи деталей, сборочные чертежи, спецификации, схемы и др. в соответствии с требованиями ЕСКД и получают начальные знания по проектированию. К сожалению, на этом знакомство с современными САПР часто заканчивается.

На третьем курсе, при выполнении курсовых проектов, от студентов требуется умение не только провести расчеты конструкций в рамках одной среды проектирования, но и перенести результаты в другие среды для проведения расчетов с последующим их возвратом в исходную среду. В условиях недостаточности знаний практического использования САПР у бакалавров многих специальностей при выполнении заданий, например, по курсам сопромата или ТММ возникают общие проблемы, связанные с визуализацией полученных результатов, анимацией процессов сборки и функционирования. К сожалению, не все студенты выполняют расчеты с использованием средств автоматизированного проектирования. Практика показывает, что средние и тяжелые САПР используют только те студенты, кто работает в СНТО, занят в НИРСах и изучают соответствующие CAD, CAE, CAM системы самостоятельно.

Мы решили выяснить, какие САПР изучают и используют будущие бакалавры. Косвенно на этот вопрос ответил конкурс работ, выполненных во время летней производственной практики студентов МГТУ им. Н. Э. Баумана на промышленных предприятиях после третьего курса. Жюри набрало два десятка работ, выполненных в САПР Компас-3D, включающих расчеты по методу конечных элементов в приложении APM FEM, анимацию работы оборудования и процессов сборки изделий. Следует обратить внимание, что все конкурсанты изучали Компас-3D самостоятельно.

По результатам конкурса можно сделать следующие выводы. Во-первых, выявлена острая потребность со стороны предприятий в получении квалифицированных кадрах, владеющих навыками работы в САПР Компас-3D российской фирмы АСКОН, особенно с учетом

введенных санкций и государственной программы, ориентированной на импорт замещение. Во-вторых, становится очевидна необходимость уже на втором курсе проводить теоретическую и практическую подготовку для решения задач проектирования, расчетов, анимации в рамках современных САПР, чтобы своевременно вооружить студентов прикладными знаниями для выполнения курсовых работ по сопромату, ТММ, спецкурсам, НИРС. Поэтому авторы выдвигают идею обучения бакалавров основам САПР на втором курсе.

Предложения по обучению бакалавров основам САПР.

Какие САПР лучше использовать в процессе обучения? Отвечая на этот вопрос, прежде всего, следует руководствоваться возможностями вуза получить ту или иную лицензию. Некоторые известные ВУЗы используют среднюю САПР SolidWorks, разработанную компанией SolidWorks Corporation, ныне являющейся независимым подразделением французской компании Dassault Systemes. В дальнейшем, естественным будет переход на тяжелую САПР CATIA той же компании. В качестве других вариантов можно предложить изучение таких систем, как T-FLEX (российская компания «Топ Системы»), охватывающая многие разделы основных программ инженерного образования, nanoCAD (российская компания «ЗАО„ Нанософт“»), аналогичная AutoCAD платформа, содержащая все необходимые инструменты базового проектирования. Последние две компании проводят дружественную политику по внедрению своих систем в учебный процесс.

С нашей точки зрения, для студентов машиностроительных специальностей целесообразно использовать российскую систему трехмерного твердотельного и поверхностного моделирования Компас-3D (российская компания «АСКОН»), интуитивно понятную и широко используемую в нашей промышленности. Это САПР среднего уровня. Одновременно с изучением Компас-3D мы рекомендуем знакомить студентов с распространенной за рубежом системой среднего уровня MCAD Solid Edge (компания «Siemens PLM Software» – подразделение департамента Industry Automation немецкого концерна Siemens AG), поддерживающую инженерный анализ деталей и сборок. Практика показывает, что начинать изучение следует с простых систем и идти к более сложным, а не наоборот. Немаловажным фактором является и то, что руководство фирм производителей указанных САПР благожелательно относится к запросам по предоставлению бесплатных лицензий для вузов.

Мы считаем, что наибольший эффект при подготовке квалифицированных бакалавров достигается за счет совмещения практических (17 ч) и лабораторных (17 ч) занятий по освоению САПР с теоретическим курсом «Основы автоматизированного проектирования», который занимает треть отведенной учебной нагрузки (17 ч). Курс планируется в составе из трех модулей:

1. Общие сведения об автоматизированном проектировании в технике.

2. Основы математического моделирования, инженерного анализа и оптимизации в САПР.

3. PDM и PLM системы в САПР.

Поскольку речь идет о студентах второго курса обучения, мы рекомендуем в лекциях рассматривать лишь общие подходы к решению задач проектирования, основное внимание, уделяя получению практических навыков работы в конкретных САПР.

Еще две трети времени обучения бакалавров САПР занимают практический (17 ч) и лабораторный (17 ч) занятия. Условно, их можно разделит на три части:

1. Изучение интерфейса и геометрическое моделирование в Компас-3D и Solid Edge. Поскольку обе системы имеют ленточный интерфейс, подобный изучаемому на первом курсе AutoCAD, он является достаточно привычным для студента. Относительно новыми являются подходы 3D-проектирования, понятия примитивов, объектно-ориентированных конструктивных элементов (**Конструктивный элемент** (КЭ, англ. **feature**) – основная единица проектирования в современных САПР, которая представляет собой элемент, задающий форму изделия. фичерсов), дерева построения, способы получения объемных тел вращением, перемещением контура, натягиванием оболочки. Особо стоит вопрос о переносе известных решений в рамках синхронного моделирования. Здесь важным фактором является потеря дерева построения, замена примитивов в исходном описании облаком точек в новой конструкции.

2. Применение расчетов предполагает использование других сред, шаблонов, ролей при использовании приложений в одной системе и перевод в определенные форматы данных в разных системах. Предполагается использовать несложные задания в двух системах, Компас-3D и Solid Edge для уточнения параметров конструкции из раздела геометрического моделирования и сопряжений сборки. Бакалавры должны получить опыт работы с программами анализа на макроуровне и микроуровне с подробным объяснением всех подготовительных действий и примером отображения полученных результатов. Результаты расчета должны отразиться в окончательной конструкторской документации на изделие.

3. Создания анимации процесса сборки спроектированного изделия, функционирования, технического обслуживания или утилизации. Любой конструктор, создающий изделие с подвижными звеньями, хочет убедиться в правильности своих расчетов до его изготовления. Ему важно отследить траектории движения деталей, проверить, не сталкиваются ли отдельные части механизма друг с другом или с окружающими элементами. Для решения всех этих задач в Компасе-3D имеется продукт – библиотека анимации, опыт работы с которой и предлагается приобрести.

На защитах квалификационных работ бакалавров следует обратить внимание на недостаточно высокий уровень представления результатов выполненных проектов. Если специалисты и инженеры на старших курсах получили опыт презентации проектов, то бакалавры его лишены. С доступной программой подготовки презентаций Microsoft PowerPoint многие знакомы еще со школьной скамьи, поэтому целесообразно использовать полученные навыки работы с ней для оформления и защиты домашних заданий, отчетов по практическим и лабораторным курсам. Таким образом, студенты могут наглядно представлять результаты своих работ и приобретать опыт в их защите.

Выводы

Бакалавры должны получить знания и навыки, необходимые для их дальнейшего роста и совершенствования на современном производстве. И, поскольку CAD/CAE/CAM системы является инструментарием геометрического, математического моделирования и представления результатов, то необходимо своевременно закладывать основу для ее грамотного использования.

Библиографический список

1. Штерензон, В. А. САПР-ориентированная подготовка бакалавров профессионального обучения / В. А. Штерензон // Современные научные исследования и инновации. – 2013. – № 10. – URL: http://web.snauka.ru/issues/2013/10/26761 (дата обращения: 19.10.2015).

2. Как преподавать САПР // isicad.ru : портал САПР, PLM и ERP. – 2011. – 13 мая. – URL: http://isicad.ru/ru/articles.php?article_num=14425 (дата обращения:19.10.2015).

М. И. Кочергин

Томский государственный университет систем управления и радиоэлектроники, Томск, Россия

КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ ПОЛЕТА ТЕЛА В АТМОСФЕРЕ ДЛЯ ОБРАЗОВАТЕЛЬНЫХ ЦЕЛЕЙ

Описывается компьютерная модель полета снаряда в атмосфере Земли, реализованная в среде моделирования MAPC, а также возможности ее использования в обучении физике и моделированию.

Задачи внешней баллистики представляют интерес для их компьютерного моделирования в образовательных целях, так как: 1) являясь динамическими системами, позволяют разъяснить суть и особенности таких систем на своем примере; 2) помимо способство-

вания углублению знаний в области физики и навыков моделирования, наглядно позволяют продемонстрировать процедуру итерационного моделирования – постепенного усложнения (детализации) модели и ее отладки на каждом этапе. Так задача о полете тела в атмосфере Земли может быть разделена на несколько уровней детализации: 1) идеальный полет тела; 2) полет тела с учетом сопротивления воздуха при его постоянной плотности; 3) полет тела с учетом изменения его аэродинамических характеристик в зависимости от его скорости; 4) полет тела с учетом изменения плотности воздуха с изменением высоты полета; 5) полет тела с учетом изменения воздействующей на него силы тяжести с изменением высоты полета; 6) полет тела с учетом кривизны Земли [1].

Наиболее подходящим для моделирования динамических систем является использование объектно-ориентированного подхода, который предполагает комплексный анализ задачи и представление ее в виде системы взаимосвязанных объектов, обладающих рядом параметров. Одним из эффективных и удобных объектно-ориентированных подходов к моделированию динамических систем, на наш взгляд, является многоуровневый подход к моделированию физико-технических систем, реализованный в среде моделирования (СМ) МАРС [2].

Формализм метода компонентных цепей. СМ МАРС базируется на методе компонентных цепей (МКЦ), основным понятием которого являются компонент и компонентная цепь (КЦ) – компонентная компьютерная модель объекта. КЦ определяется как совокупность объектов $C = (K, B, N)$, где K – множество компонентов цепи; B – множество ветвей цепи; N – множество узлов цепи. Каждый компонент имеет произвольное число связей (информационного, элементарного или векторного типа) и относится к одному из типов: *компоненты-источники* данных и энергии, *компоненты-преобразователи* и *компоненты-измерители*. Узлы компонента, представляющие собой общую точку соединения полусвязей нескольких связанных компонентов, служат для передачи информационных данных.

Многоуровневое представление компьютерных моделей. На рис. 1 представлена общая структура многоуровневого (многослойного) представления [3] компьютерной модели в рамках формализма МКЦ.



Рис. 1. Структура многоуровневого представления модели

Таким образом любая компьютерная модель в СМ МАРС может быть представлена на трех слоях: визуальном, логическом и объектном. Такое разделение позволяет отделить модель непрерывного (физического) поведения объекта от дискретного (логического – алгоритма поведения объекта или сценария проведения виртуального эксперимента), а также от процесса ввода/вывода данных пользователю (интерфейса).

Визуальный слой (V-слой) предназначен для ввода значений входных данных модели, вывода пользователю результатов измерений выходных данных и визуализации результатов моделирования в виде графиков, диаграмм или с использованием графической анимации.

Объектный слой (С-слой) содержит математическую модель системы или объекта в виде КЦ, характеризующей их непрерывное (в физико-технических задачах – физическое) поведение.

Логический слой (L-слой) осуществляет передачу данных между объектным и визуальным слоями, а также содержит алгоритмическую КЦ, определяющую сценарий проведения эксперимента над моделью или же алгоритм дискретного (логического) поведения моделируемого объекта.

Такое разделение специфицирует и упрощает процедуру внесения изменений в модель, так, например, добавление дискретных ограничений на модель влечет ее изменение только на логическом слое, а детализация модели может потребовать внесения изменений либо только на объектном слое, либо на объектном и логическом (в зависимости от характера усложнения модели).

Многоуровневая компьютерная модель полета тела в СМ МАРС. Рассмотрим представление второго уровня детализации описанной задачи о полете тела в атмосфере Земли – полет тела с учетом сопротивления воздуха при его постоянной плотности. Данная задача содержит два активных объекта: «Тело» и «Атмосфера». На рис. 2 представлена многоуровневая компьютерная модель данной задачи в СМ МАРС.

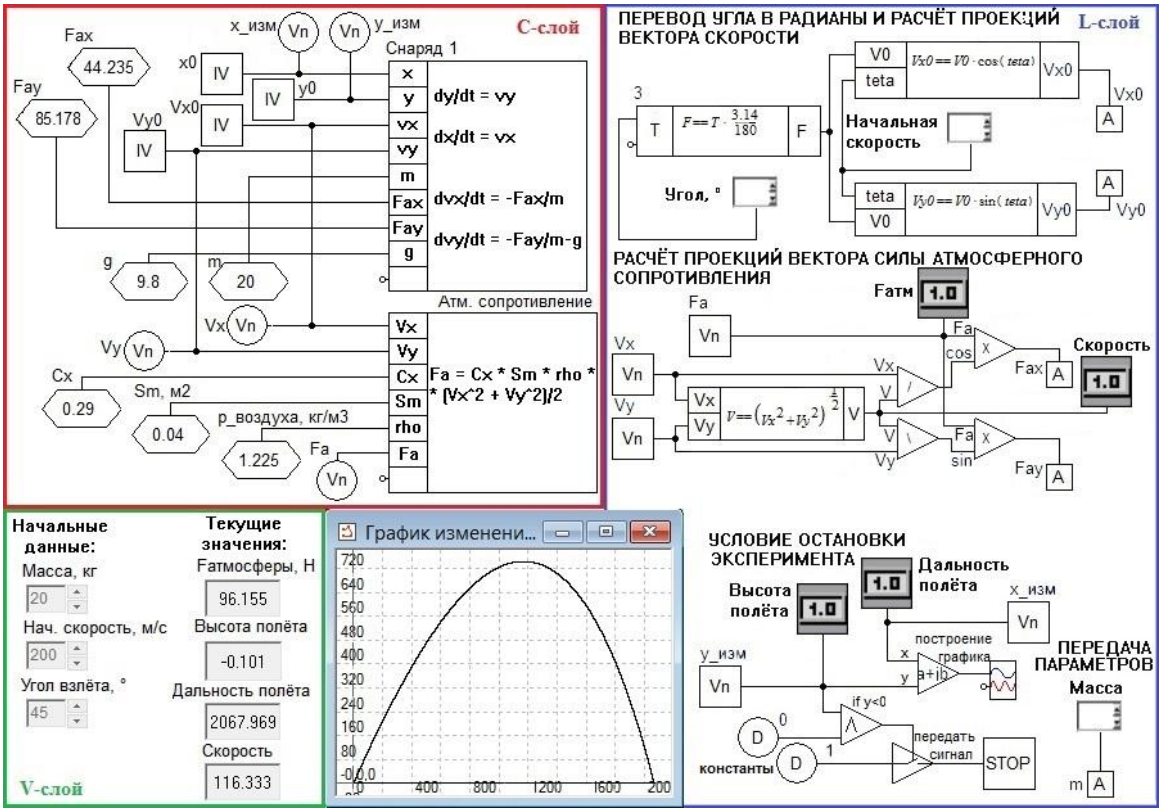


Рис. 2. Многоуровневая компьютерная модель полета тела

На *визуальном слое* расположены три компонента «Цифровое табло со спином» для ввода начальных данных (массы, начальной скорости и начального угла полета тела) и четыре компонента «Цифровое табло» для вывода результатов расчета модели (текущие значения силы атмосферного сопротивления, высоты и дальности полета, скорости). Также в отдельном окне отображается график изменения координат полета тела.

На *объектном слое* расположена КЦ, описывающая непрерывное физическое поведение тела, совершающего полет в атмосфере Земли, которая состоит из 1) двух *интерактивных математических панелей* (ИМП) – «Снаряд 1» и «Атм. сопротивление», 2) *источников начального значения переменной* (изображены в виде квадратов с надписью *IV* внутри: «x0», «y0», «Vx0», «Vy0»), 3) *измерителей переменных* (изображены в виде

окружностей с надписью *Vn* внутри: «x_изм», «y_изм», «Vx», «Vy», «Fa»), 4) *источников физической величины*, которые также имеют одноименные отображения на логическом слое («Fax», «Fay», «g», «m», «Cx», «Sm», «p_воздуха, кг/м3»). ИМП «Снаряд 1» характеризует непрерывное поведение объекта «Тело» и содержит систему дифференциальных уравнений первого порядка описывающих траекторию полета тела массой *m*, запущенного под углом θ с начальной скоростью V_0. ИМП «Атм. сопротивление» характеризует объект «Атмосфера» и описывает зависимость силы атмосферного сопротивления от скорости тела и его аэродинамических свойств по формуле $F_a = C_X \cdot S_M \cdot \frac{\rho \cdot V^2}{2}$, где C_X – безразмерный коэффициент силы аэродинамического лобового сопротивления (в данной модели является константой и равен 0,29), S_M – геометрическая характеристика движущегося объекта (площадью «миделевого сечения» – в данной модели равняется 0,04 м^2), ρ – плотность воздуха (для данной модели взято значение плотности воздуха на уровне моря при температуре 15 °C равное 1,225 кг/м^3). *Компоненты-измерители переменных* передают рассчитанное значение на логический слой, а те *компоненты-источники физической величины*, которые имеют одноименное отображение на логическом слое (в виде квадрата с надписью *A* внутри), передают значения переменной с логического слоя на объектный.

На логическом слое расположена КЦ, реализующая дискретное событие – завершение эксперимента при выполнении условия $y_{\text{текущее}} < 0$, а также две КЦ реализующие вспомогательные расчеты: перевод величины угла в радианы и расчет проекций вектора скорости и атмосферного сопротивления. Эти КЦ состоят из 1) отображений *компонентов-измерителей* объектного слоя, выступающих на этом слое в качестве источников (обозначены в виде квадратов с надписью *Vn* внутри: «x_изм», «y_изм», «Vx», «Vy», «Fa»), 2) четырех ИМП, содержащих формулы для расчета, 3) компонентов-измерителей логического слоя («Fатм», «Скорость», «Высота полета», «Дальность полета»), передающих данные на одноименные цифровые табло визуального слоя, 4) других общеалгоритмических компонентов. Также логический слой выступает в качестве промежуточного звена для передачи данных из визуального слоя в объектный и обратно.

Использование компьютерной модели в образовательных целях. Работа с такими многоуровневыми моделями может преследовать цель как углубления знаний в области физики, так и закрепления навыков моделирования.

В первом случае модель выступает в качестве инструмента обучения, позволяющего проводить вычислительный эксперимент и визуализировать результаты моделирования. Для этих целей СМ МАРС позволяет преобразовать любую созданную в ней модель в отдельное исполняемое приложение – учебно-иллюстративный модуль (УИМ) [4], который может использоваться в качестве дополнения к электронному учебному пособию или являться частью электронного курса, созданного, например, в системе дистанционного образования Moodle (в качестве иллюстративного материала к лекции или элемента домашнего, лабораторного занятия). Используя УИМ, студенты могут исследовать поведение моделируемого объекта, выявлять зависимость между различными величинами и оценивать то, какое влияние на поведение объекта оказывает изменение того или иного параметра модели. Также студентам может быть предложено сравнение и сопоставление результатов виртуальных экспериментов, полученных на моделях различной степени детализации.

Во втором случае компьютерная модель выступает объектом изучения: студенту необходимо взаимодействовать с самой средой моделирования –решать различные задачи связанные с детализацией модели или введением каких-либо ограничений или дополнительных условий. В качестве критериев оценивания решения поставленных задач, выполненного пользователем на компьютерной модели, можно использовать следующие: адекватность составленной модели, корректность составленного алгоритма поведения объекта, правильность составления модели, обоснованность выбора компонентов (достаточность и неизбыточность состава КЦ).

Заключение. Использование инструментальных средств, таких как СМ МАРС, предоставляет возможность быстрой разработки компьютерных моделей с высокой степе-

нью визуализации и проведения виртуальных экспериментов, не требующих написания программного кода на языках программирования. Использование таких инструментальных средств на занятиях различного вида и во время самостоятельной работы студентов позволяет повысить степень их вовлеченности в работу, интерактивности и наглядности, а как следствие, улучшить качество преподавания и результаты учебной деятельности [5].

Библиографический список

1. Колесов, Ю. Б. Моделирование систем. Практикум по компьютерному моделированию / Ю. Б. Колесов, Ю. Б. Сениченков. – СПб. : БХВ-Петербург, 2007. – 352 с.

2. Дмитриев, В. М. Система визуализации и управления вычислительным экспериментом в среде многоуровневого моделирования МАРС / В. М. Дмитриев, Т. В. Ганджа, Т. Ю. Коротина // Доклады ТУСУР. – 2010. – Т. 1, № 2. – С. 149–155.

3. Дмитриев, В. М. Многоуровневое моделирование задач физики / В. М. Дмитриев, Т. В. Ганджа, М. И. Кочергин // Современное образование: практико-ориентированные технологии подготовки инженерных кадров : материалы Междунар. науч.-метод. конф. (29–30 января 2015 г., Томск). – Томск : Изд-во Томск. гос. ун-та систем упр. и радиоэлектроники, 2015. – С. 47–49.

4. Дмитриев, В. М. Формирование учебно-иллюстративных модулей в среде многоуровневого компьютерного моделирования МАРС / В. М. Дмитриев, Т. В. Ганджа // Электронные средства и системы управления. – 2013. – № 2. – С. 96–100.

5. Королев, А. Л. Компьютерное моделирование в образовании / А. Л. Королев // Problems of modern pedagogics in the context of international educational standards development : Materials digest of the XL International Research and Practice Conference and I stage of the Championship in Pedagogical sciences (London, January 31 – February 05, 2013). – London : IASHE, 2013. – С. 126–128.

Ш. Л. Сулипов, Х. Б. Шахмерзаева, Л. И. Ламаева

Чеченский государственный педагогический университет, Грозный, Россия

ОРГАНИЗАЦИЯ ПРОЕКТНО-ИССЛЕДОВАТЕЛЬСКОЙ ДЕЯТЕЛЬНОСТИ СТУДЕНТОВ НА УРОКАХ ИНФОРМАТИКИ

Рассматриваются вопрос об организации проектно-исследовательской деятельности студентов на уроках информатики, метод проекта, использование электронных учебников.

Метод образовательного проекта основан на методе применение информационных-коммуникационных технологии. Это одна из личностно ориентированных технологий, способ организации самостоятельной деятельности студентов, направленных на решение проблем образовательного проекта, интеграцию проблемного подхода, группового, рефлексивного, презентационного, исследовательского, поискового и других методов.

Образовательный проект с точки зрения ученика – это возможность сделать что-то интересное самостоятельно, в группе или себе, максимально используя свои возможности; это деятельность, которая позволяет проявить себя, попробовать свои силы, применить свои знания; это деятельность, направленная на решение интересной проблемы, сформулированной самими учениками в виде цели и задачи, когда результат этой деятельности – метод, найденный для решения проблемы – имеет практический характер, имеющий большое практическое значение и, что очень важно, интересно и важно для самих первооткрывателей.

Образовательный проект с точки зрения учителя – это дидактические средства развития, образования и воспитания.

Цель – преподавание студентов в уроках информатики по методу проектов, как способ обучения, – может быть полностью достигнуто путем изучения программы Microsoft PowerPoint.

Исследование – один из видов человеческой когнитивной деятельности, установление, обнаружение, понимание реальности, приобретение новых знаний.

В отличие от научных исследований, основной целью которых является получение объективно новых знаний, студенты в ходе исследований получают субъективно новые знания (новые и лично значимые для конкретного студента). В то же время наблюдается повышение мотивации к учебной деятельности и активизации личной позиции ученика в учебном процессе. Целью исследовательской деятельности в области образования является получение студентами функционального навыка исследования как универсального способа овладения реальностью.

Обычно принято определять следующие пути и методы исследования:

• возможность видеть проблемы;

• способность разрабатывать гипотезы;

• способность наблюдать;

• возможность проведения экспериментов;

• возможность определять понятия и т.д.

Одним из способов активизации познавательной деятельности студентов в уроках информатики является использование электронных учебников на уроках.

Применение электронного учебника на уроках информатики позволяет решать следующие задачи:

• повысить когнитивный интерес студентов к предмету «Информатика»;

• создание условий для овладения знаниями, умениями, и навыками учащихся;

• работа с интернет-ресурсами, поиск соответствующей информации в Интернете, которые необходимы современному человеку для дальнейшего обучения и развития;

• формирование учебного процесса по принципу «ученик – предмет обучения»;

• моделирование поисковой деятельности студентов.

Существует общепринятый тезис о том, что образовательный проект должен стать основной формой образовательной деятельности в условиях обучения в старших классах, а использование ИКТ для реализации проектов является обязательным и наиболее эффективным способом его реализации. В то же время выделяют две цели в области образования:

– реализация модели профессиональной деятельности (деловой игры) школьника в рамках определенной образовательной области, выбранной в качестве профиля обучения;

– формирование информационной и коммуникационной компетентности учащегося путем целенаправленного использования знаний и навыков, полученных в уроках информатики, а также их роста в процессе проектной деятельности.

Модель профессиональной деятельности должна проводиться в самых современных формах, в условиях, близких к формам, внедряемым в процессе информатизации практически всех видов профессий. Это означает, что все этапы работы проекта (независимо от предметной области) должны быть реализованы с использованием инструментов ИКТ. Такие этапы включают:

1) постановка задачи; 2) подготовка технической задачи (ТЗ) для разработки проекта; 3) планирование реализации проекта; 4) сбор информации о проекте; 5) обработка собранной информации; 6) отчет о результатах проекта; 7) презентация проекта.

Другим важным моментом проектной работы школьников является организация

На втором этапе организована учебная деятельность. Если проект представляет собой групповой проект, то необходимо организовать детей в группы, определить цели и задачи каждой группы. При необходимости определите роль каждого члена группы. На этом же этапе происходит планирование работы по решению проектной задачи. Он может быть параллельным или последовательным.

Вывод. Таким образом, можно сделать вывод о том, что необходимость применения методологии исследования исследований в современном школьном образовании обусловлена очевидными тенденциями в системе образования для более полного развития личности студента, его подготовки к реальной деятельности. Наши собственные наблюдения показали, что в целом такой метод является эффективной инновационной технологией, которая значительно повышает уровень компьютерной грамотности, внутреннюю мотивацию учеников, уровень независимости студентов, их терпимость и общее интеллектуальное развитие.

П. К. Чечель, А. А. Краснов, Д. В. Пащенко

Пензенский государственный университет, Пенза, Россия

СРАВНИТЕЛЬНЫЙ АНАЛИЗ ОРГАНИЗАЦИИ УЧЕБНОГО ПРОЦЕССА В ВЫСШИХ УЧЕБНЫХ ЗАВЕДЕНИЯХ РОССИИ И ЕВРОПЫ. РАЗНИЦА МЕТОДОВ И ТЕХНОЛОГИЙ

Рассматривается опыт обучения в высшем учебном заведении Европы по технической специальности. Представлен сравнительный анализ современных систем высшего образования в России и Европе.

В данной статье рассмотрены примеры и сравнения особенностей организации учебного процесса в главном региональном Российском ВУЗе и главном университете Румынского жудеца (административно-территориальная единица Румынии), на основе полученного опыта очного обучения за рубежом в 2016\2017 учебном году.

Несмотря на то, что в Российском ВУЗе насчитывается более двадцати тысяч студентов, а в Румынском университете не более двух с половиной тысяч – уровень учебных программ и глубина приобретаемых знаний находятся примерно на одном уровне.

Румынская система образования на всех ее ступенях соответствует Болонской системе. Хотя с 2003 г. Россия выступает членом Болонской системы образования, после многочисленных реформ сферы образования процесс обучения в Российских вузах отличается от аналогичного процесса в Европе. Данные различия присутствуют на всех уровнях системы образования.

Так Европейская программа бакалавриата подразумевает собой три года обучения, против четырех в РФ. Отличаются и принципы защиты предмета. Преподаватель на свое усмотрения задает форму экзамена. Так, чтобы получить отметку и закрыть курс, студент должен разработать и защитить проект, тематика которого завязана на теоретической составляющей курса лекций. Таким образом, защита проекта является важной частью процесса получения новых навыков, так как данный вид деятельности направлен на самостоятельную работу студентов, а также, в зависимости от успешности защиты и презентации проекта, студенты получают оценку за данный учебный курс.

Каждое занятие включает в себя, следующие последовательные части. Проверка домашней работы занимает первые 10–20 мин занятия. Домашнее задание отправляется по электронной почте преподавателю для предварительной проверки минимум за день до занятия – это экономит время, затрачиваемое на проверку работ студентов непосредственно на занятии. Во время проверки преподаватель лишь указывает на плюсы и недочеты домашней работы и отвечает на возникшие в ходе ее выполнения вопросы. Затем преподаватель рассказывает новый материал сопровождая его примерами. Студенты, как правило, делают пометки, но не в виде конспектов лекций, а как памятку с самыми интересными и ключевыми схемами, примерами, таблицами и определениями. Такие записи слабо структурированы, но при этом отлично помогают сохранить в памяти основные положения курса. Наличие или отсутствие записей теоретического материала у студента никак не влияет на итоговую оценку. Также часто преподаватель сам раздает лекционный материал студентам, рассылая его посредством электронной почты или организовывая общий доступ к папке с материалами в облачной хранилище.

Завершающим этапом занятия является закрепление и апробация изученной темы на практических упражнениях. Во время выполнения последних, студент может спросить у преподавателя непонятные ему моменты или задать дополнительный вопрос касательно рассматриваемой на занятии темы. За десять минут до окончания занятия студентам выдается новый набор заданий для самостоятельной работы и освоения учебного материала, а также рекомендуются к изучению дополнительные источники: будь то книги, электронные ресурсы или иные формы информации.

Более половины учебной нагрузки студента, как правило, приходится на подготовку домашнего задания и самостоятельное углубленное изучение темы. При этом преподавателю не составит труда определить, насколько хорошо студент подготовился к занятию, ведь при обсуждении новой темы видно, насколько студентом хорошо была изучена предыдущая тема.

Каждый курс имеет четко прописанный и утвержденный список тем. Тем не менее, преподаватель решает совместно со студентами, какие темы разобрать подробнее, а какие в общих чертах. Диалог преподавателя и студента присутствует и активно ведется на протяжении всего учебного процесса и иногда даже за его пределами. На фоне этих обстоятельств студенты чувствуют себя более вовлеченными в процесс учебы.

Иначе в Европейских университетах относятся и к вопросу посещаемости. Если студент по каким-либо причинам не может присутствовать на занятии, он заранее пишет преподавателю, что на следующем занятии его не будет и указывает причину пропуска занятия. Проблемы со здоровьем, накладка в собственном расписании или даже форс-мажорные обстоятельства – студенты уведомляют преподавателя об отсутствии. В противном случае, у студента может состояться неприятный разговор в деканате своего факультета, даже после единичного пропуска. Естественно, что право на тайну частной жизни и тайну медицинской истории никто не стремиться нарушить: студенту достаточно сослаться на здоровье – не важно болен он или идет на профилактический прием к врачу. В свою очередь преподаватели также стремятся уведомить своих студентов о том, что следующее занятие по тем или иным причинам откладывается. Данная деталь дополнят картину доверия и взаимоуважения студентов и преподавателей.

Сайты заведений высшего Европейского образования имеют специальный легкодоступный раздел с расписаниями преподавателей. Расписание любого преподавателя можно узнать на официальном сайте университета, а также в университетском приложении, написанном под все популярные мобильные платформы. В расписании отображаются не только занятия, но и дополнительные курсы, мероприятия и экзаменационные встречи. Данная прозрачная система расписаний значительно упрощает поиск сотрудников университета со стороны студентов. На ряду с этим, существуют и расписания для групп, но такие расписания не публикуются на сайтах в свободном доступе, так как ущемляют права на тайну частной жизни студентов. Безусловно, расписание группы могут увидеть представители университета, но для третьих лиц данная информация закрыта.

Понятие единой нормы и объема семестровых работ отсутствует. Преподаватель может корректировать объем и сложность работ для отдельных студентов. Так, когда преподаватель замечает отставание студента по предмету, либо его низкие компетенции в этой предметной области – домашние задания, описанные выше, и объем литературы для самостоятельного изучения, уменьшается. Наоборот, если некоторые студенты идут с опережением, отлично разбираются в материале и с легкостью выполняют все задания – им предлагаются к решению более объемные сложные задачи, иногда требующие нестандартного решения, дополнительные материалы, написанные более сложным и профессиональным языком. Данная корректировка проходит во время семестра и может быть проведена несколько раз к ряду. Такой индивидуальный подход влияет на итоговый балл, ведь выполнение облегченных заданий и пропуск нескольких заданий не может быть приравнен к полному комплексному изучению курса. Ситуация обратна при усилении учебной программы – ведь студент, блестяще освоивший больше материала, чем его группа, должен получить за курс максимальный балл. Таким образом достигается балансировка нагрузки и индивидуальный подход к каждому студенту: слабые студенты не чувствует себя перегруженными и отстающими, а для одаренных же студентов появляется больше возможностей попрактиковаться и отточить свои навыки.

На сессиях, помимо проектов, весьма распространены письменные экзамены. Сдавая такой экзамен, студент должен за строго оговоренное и ограниченное время письменно выполнить задания, выдаваемые каждому студенту индивидуально и в случайном порядке. Если это экзамен по техническому или IT-курсу, то письменный экзамен может перекочевать в электронную плоскость (требуется написать программу, скрипт, алгоритм и т.д.). Результат такого экзамена не объявляется сразу, и студенты вынуждены ждать следующей встречи с преподавателем для оглашения результатов и их фиксации в студенческих книжках.

Кроме вышеперечисленных способов защиты предмета, существует масса других, будь то устный экзамен, сумма всех домашних и самостоятельных работ, блиц-опрос. Часто защита предмета принимает вид уникальной комбинации методов, позволяющей максимально раскрыть и определить уровень знания студентов.

Преподаватели помимо непосредственно образовательной деятельности участвуют в постоянно функционирующих программах обмена кадрами между университетами. Являются активными участниками мастер-классов, научных лабораторий и конференций. Научная жизнь преподавательского состава насыщена международными выставками, после посещения которых, новые технологии и методики внедряются в родных университетах. Поэтому подавляющее число высших учебных заведений Европы обладают сравнительно одинаковым уровнем качества образования.

По-иному устроен процесс выпуска студентов. Этот процесс схож с Российским выпуском учеников из школ. В конце мая студенты отмечают выпускной. Наряжаются в мантии и костюмы бакалавров, слушают выпускные речи с напутствиями, устраивают празднества, фотосессии и принимают поздравления от друзей и близких. На выпускных мероприятиях, студентам вручают лишь некий эквивалент благодарственного письма, где говорится, что университет рад что обучал такого замечательного студента, а ныне выпускника. Диплом и приложение к диплому студенты получают лишь после закрытия последней сессии и успешной защиты своей дипломной работы. Получив все документы, свидетельствующие о наличии высшего образования бывшие студенты в большинстве перебираются в крупные города с бóльшим количеством вакансий, соответствующих их профессиональному образованию.

Таким образом, имея многочисленные различия, Западная и Российская системы высшего образования имеют также и много общего. Имея, колоссальную разницу в финансировании, обе системы имеют ряд общих проблем, связанных в первую очередь с потерей актуальности некоторых предметов. Но, так как на территории Европейского пространства университеты тесно связаны между собой партнерскими отношениями, замена устаревших методик и курсов происходит в разы быстрее благодаря тому, что университеты для разработки и внедрения нового курса часто прибегают к совместной работе. Более полная и глубокая интеграция Российской системы с Европейской системой образования в перспективе должна повысить конкурентоспособность и престиж Российского образования, а также поспособствовать всеобщей актуализации предметных областей и учебных курсов, входящий в эти области. В интеграции присутствует и еще один плюс, касающийся либерализации учебного процесса и укрепления связей, возникающих в ходе процесса обучения между студентов и преподавателем.



Рис. 1. Фото с представителями международного офиса на фоне главного корпуса университета в Румынии

Библиографический список

1. Лашко, С. И. Модернизация высшего образования в Европе и России / С. И. Лашко, Л. В. Левченко // Экономика образования. – 2008. – № 1. – С. 109–114.
2. Краснова, О. В. Механизм развития систем педагогических взаимодействий как психолого-педагогическая основа проектирования электронных систем обучения / О. В. Краснова, А. С. Мещеряков, С. А. Влазнева // Современные проблемы науки и образования. – 2014. – № 3. – С. 187.
3. Краснова, О. В. Развитие информационно-технологической компетентности в вузе на основе теории функционирования и развития систем педагогических взаимодействий / О. В. Краснова, А. А. Краснов // Педагогическая информатика. – 2013. – № 3. – С. 21–29.

Е. С. Брусникин, В. И. Кичук, Н. Н. Коннов

Пензенский государственный университет, Пенза, Россия

ГРАФИЧЕСКИЙ РЕДАКТОР ДИАГРАММ МИКРОПРОГРАММНОЙ ЛОГИКИ

Приводится описание программы, предназначенной для автоматизации проектирования микропрограммируемых средств ВТ в учебном процессе подготовки бакалавров по направлению «Информатика и вычислительная техника».

Подготовка современного специалиста в области вычислительной техники предусматривает знакомство с организацией ЭВМ на микропрограммном уровне. Любая машинная команда исполняется аппаратурой не непосредственно, а путем их интерпретации в соответственную последовательность более простых действий – межрегистровых передач. А значит, всегда существует задача программирования машинных команд из более простых действий – микропрограммирование [1].

Микропрограммный уровень в современных процессорах скрыт внутри кристалла процессора, что не позволяет изучать его на реальных ЭВМ. Поэтому для изучения микропрограммирования и освоения навыков проектирования микропрограмм широко используются программные эмуляторы. На кафедре вычислительной техники ПГУ были разработаны и активно применяются в лабораторном практикуме и курсовом проектировании эмулятор шестнадцатиразрядного микропрограммируемого процессора, использующего логическую организацию некогда популярного разрядно-модульного комплекта серии К1804 [2, 3].

Разработка и отладка микропрограмм процесс весьма трудоемкий, что затрудняет ручное кодирование микропрограмм. Средством описания работы микропрограммируемых устройств являются диаграммы микропрограммной логики (ДМЛ), которые позволяют описать каждую микрокоманду на уровне регистровых передач и взаимодействие микрокоманд. ДМЛ представляет собой соединение блоков, каждый из которых является подробным описанием одной микрокоманды. Изображение блоков стандартизовано.

Для автоматизации подготовки ДМЛ и процесса переноса логического описания работы программы в микропрограммный код был создан специализированный графический редактор, работающий в среде Microsoft Visio 7/10/12. Редактор обеспечивает документирование микропрограмм для для учебной микроэвм с учетом всех требований к оформлению, а также непосредственно генерирует код микропрограммы, который пользователь затем переносит в ПЗУ или в эмулятор. Редактор представляет собой набор встроенных в шаблон документа Visio макросов, форм и библиотеку фигур на языке VBA (Visual Basic for Applications) [4].

Однако принятые правительственные решения о переходе органов власти и бюджетных учреждений на использование свободного программного обеспечения заставляет вести работу по замене ПО Microsoft, используемого в учебном процессе, на отечественное либо свободно распространяемое ПО.

Авторами была разработана новая программа графического редактора ДМЛ, которая осуществляет ввод мнемонических обозначений полей микрокоманды данных через интуитивно понятный графический интерфейс, осуществляет их проверку на корректность, обеспечивает хранение непосредственно в графическом документе, а также выгрузку данных в виде таблицы прошивки ПЗУ в файл, совместимый с форматом учебного эмулятора процессора.

Основные особенности программы:

- реализована с помощью макросов в среде OpenOffice Draw, которая является свободно распространяемым продуктом;
- программа кроссплатформенная и может быть использована на популярных ОС таких как Linux/Windwos/Mac OS;
- диалог настроек сделан с помощью наглядных элементов управления, уменьшающих вероятность совершения оператором ошибки.

Программа написана на язык макросов OpenOffice Basic, который был специально разработан для OpenOffice.org и сильно интегрирован в офисный пакет. OpenOffice Basic является диалектом Visual Basic for Applications и имеет существенных похожие синтаксис и в семантику, что позволяет его код легко экспортировать на изделия компании Microsoft. Основным отличием OpenOffice Basic от MS VBA является Java подобная классификация объектов [5, 6].

Программа разделена на три модуля внутри одного файла.

Первый модуль содержит функции работы с диалоговым окном, а также функции по добавлению, считыванию и выводу данных. Одни из них это заполнение объектов диалогового окна (FillListBox, FillTextBox, FillBits, FillRadio), добавление и считывание параметров-данных из/в диаграмму (getUserProperty, addUserProperty), вывод самого диалогового окна (ShowDialog), а также заполнение диаграммы мнемоникой микрокоманды после сохранения и закрытия диалогового окна (SetShapeText, SaveAndCloseDialog).

Второй модуль отвечает за информацию, содержащуюся в объектах диалогового окна, а также их краткое представление для отображения на диаграмме.

Третий модуль программы отвечает за экспорт данных в формат эмулятора PZU. Он содержит функцию транслятора введенных данных пользователем в двоичный код уже читаемый эмулятором.

Пример рабочего окна графического редактора приведен на рис. 1.

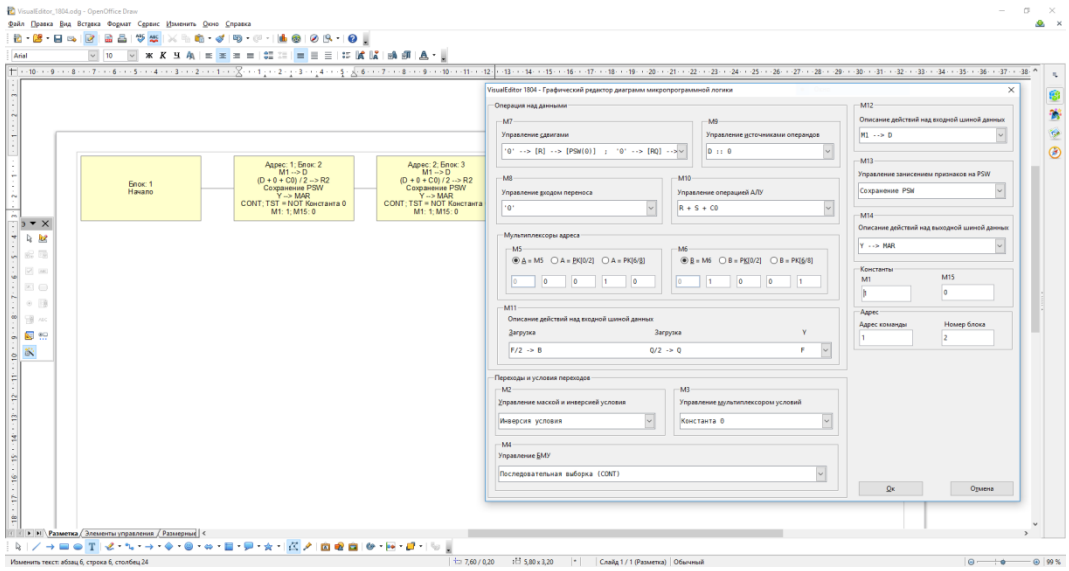


Рис. 1. Окна графического редактора

Текст на блоках в диаграмме отображает мнемонику микрокоманды:

- адрес команды и номер блока;
- источники и приемники данных входной шины;
- операция алу с указанием аргументов и местом загрузки результата;
- управление содержимым регистра состояния PSW;
- приемник данных выходной шины;
- переход и условие для его совершения;
- константы.

Константы и адреса микрокоманд задаются числами в восьмеричной системе счисления.

Созданный документ OpenOffice Draw с разработанными макросами будет работать в свободно распространяемом пакете FreeOffice.

Разработанный графический редактор диаграмм микропрограммой логики будет использован в лабораторном практикуме по дисциплине «ЭВМ и периферийные устройства».

Библиографический список

1. Таненбаум, Э. Архитектура компьютера / Э. Таненбаум. – СПб. : Питер, 2003. – С. 695.

2. Коннов, Н. Модернизация учебного программного обеспечения по курсу ЭВМ и периферийные устройства / Н. Н. Коннов, А. П. Любезнов, М. Ю. Новиков, И. В. Янович // Информационные технологии в науке и образовании. Проблемы и перспективы : сб. науч. ст. III ежегодной межвуз. студ. науч.-практ. конф. – Пенза : Изд-во ПГУ, 2016. – С. 72–73.

3. Коннов, Н. Учебный программный эмулятор ЭВМ с микропрограммным управлением / Н. Н. Коннов, С. Е. Филин, П. П. Вирясов, К. О. Барабаш // Университетское образование (МКУО–2014) : сб. ст. XVIII Междунар. науч.-метод. конф., посвящ. 200-летию со дня рождения М. Ю. Лермонтова (г. Пенза, 10–11 апреля 2014 г.) / под ред. А. Д. Гулякова, Р. М. Печерской. – Пенза : Изд-во ПГУ, 2014. – С. 183–184.

4. Захаров, С. А. Свидетельство о гос. регистрации программ для ЭВМ. Графический редактор диаграмм микропрограммной логики / С. А. Захаров, А. П. Захаров, Н. Н. Коннов. – 2013. – № 2013618806.

5. Козодаев, Р. OpenOffice.org 3. Полное руководство пользователя – (Библиотека ГНУ / Линуксцентра) / Р. Ю. Козодаев, А. В. Маджугин ; под ред. Е. В. Ушаковой. – СПб. : БХВ-Петербург, 2010. – С. 704.

СОДЕРЖАНИЕ

АППАРАТНО-ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ИНФОРМАЦИОННО-ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ И СИСТЕМ УПРАВЛЕНИЯ

Дроздов Д. Н., Патил С., Дубинин В. Н.
ФОРМАЛЬНАЯ ВЕРИФИКАЦИЯ КИБЕРФИЗИЧЕСКИХ СИСТЕМ УПРАВЛЕНИЯ СО СЛУЧАЙНЫМИ ВРЕМЕННЫМИ ЗАДЕРЖКАМИ 3

Дроздов Д. Н., Патил С.
ПОСТРОЕНИЕ РАСПРЕДЕЛЕННЫХ СИСТЕМ УПРАВЛЕНИЯ ПРОМЫШЛЕННОЙ АВТОМАТИКОЙ НА ОСНОВЕ СТАНДАРТА IEC 61499: ДЕКОМПОЗИЦИЯ И ОТЛАДКА 6

Дроздов Д. Н., Патил С.
РАСПРЕДЕЛЕННАЯ СИСТЕМА УПРАВЛЕНИЯ ПРОИЗВОДСТВЕННОЙ ЛИНИЕЙ FESTO MPS-500 9

Бикташев Р. А., Вашкевич Н. П.
СТРУКТУРНАЯ РЕАЛИЗАЦИЯ И МОДЕЛИРОВАНИЕ АЛГОРИТМОВ ПЛАНИРОВАНИЯ/ДИСПЕТЧЕРИЗАЦИИ В ПАРАЛЛЕЛЬНЫХ СИСТЕМАХ НА ОСНОВЕ ЛОГИКИ НЕДЕТЕРМИНИРОВАННЫХ АВТОМАТОВ 12

Бикташев Р. А.
МОДЕЛЬ УПРАВЛЕНИЯ ПРОЦЕССАМИ В МНОГОПРОЦЕССОРНЫХ СИСТЕМАХ НА ОСНОВЕ СТОХАСТИЧЕСКИХ СЕТЕЙ МАССОВОГО ОБСЛУЖИВАНИЯ 17

Федюнин Р. Н.
СПЕЦИАЛИЗИРОВАННЫЕ АРИФМЕТИКО-ЛОГИЧЕСКИЕ УСТРОЙСТВА НА БАЗЕ КОНВЕЙЕРНЫХ МАТРИЧНЫХ ВЫЧИСЛИТЕЛЕЙ 20

Беззатеев И. А., Дубинин В. Н., Уваров Д. А., Вяткин В. В.
РЕАЛИЗАЦИЯ РАСПРЕДЕЛЕННЫХ СЕМАФОРОВ В АРХИТЕКТУРЕ IEC 61499 НА ОСНОВЕ АЛГОРИТМОВ ДЛЯ РЕШЕНИЯ ЗАДАЧ КОНСЕНСУСА В СЕТИ НЕНАДЕЖНЫХ ПРОЦЕССОВ 25

Сафронов Д. В., Дубравин А. В.
АВТОМАТИЗАЦИЯ ПОИСКА ИЗБЫТОЧНЫХ ФУНКЦИОНАЛЬНЫХ ОБЪЕКТОВ В ИСПОЛНИМЫХ ФАЙЛАХ ДЛЯ ПЛАТФОРМЫ MICROSOFT.NET 29

Вашкевич Н. П., Мясин П. Ю., Поздняков С. Ю.
СТРУКТУРА МНОГОПРОЦЕССОРНОЙ СИСТЕМЫ С АППАРАТНЫМ ПЛАНИРОВЩИКОМ 31

Калачев А. В.
РАЗРАБОТКА БАЗОВОГО ФУНКЦИОНАЛЬНОГО БЛОКА УПРАВЛЕНИЯ ДЛЯ СИСТЕМЫ ПЕРЕМЕЩЕНИЯ БАГАЖА В СИСТЕМЕ NXTCONTROL 34

Уваров Д. А.
ВСТРАИВАНИЕ БАЗ ЗНАНИЙ В SOA 38

Беззатеев И. А., Дубинин В. Н., Уваров Д. А.
МОДЕЛИРОВАНИЕ РАСПРЕДЕЛЕННЫХ СЕМАФОРОВ НА ОСНОВЕ ПРОТОКОЛОВ PAXOS И RAFT В СИСТЕМЕ CPN TOOLS 41

Сенокосов И. В.
ФУНКЦИОНАЛЬНО-БЛОЧНАЯ РЕАЛИЗАЦИЯ МАГАЗИННЫХ АВТОМАТОВ 45

Бождай А. С., Евсеева Ю. И., Гудков А. А.
ОПТИМИЗАЦИЯ ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА АДАПТИВНЫХ СИСТЕМ С ПОМОЩЬЮ ТЕХНОЛОГИИ ИНТЕЛЛЕКТУАЛЬНОГО АНАЛИЗА ДАННЫХ 49

Евсеева Ю. И.
МНОГОУРОВНЕВАЯ АРХИТЕКТУРА АДАПТИВНОЙ
ИНФОРМАЦИОННОЙ СИСТЕМЫ 52

Балашова И. Ю., Шибанов С. В., Афонин А. Ю.
ПРИМЕНЕНИЕ ТЕХНОЛОГИИ БЛОКЧЕЙН В ПОСТРОЕНИИ
ДЕЦЕНТРАЛИЗОВАННЫХ РАСПРЕДЕЛЕННЫХ ИНФОРМАЦИОННЫХ СИСТЕМ 54

Ломизов Д. В., Дубравин А. В.
АВТОМАТИЗАЦИЯ ТЕСТИРОВАНИЯ ПРОГРАММНО-АППАРАТНЫХ
КОМПЛЕКСОВ 57

Дорофеева О. С, Казаков Б. В., Казакова И. А.
ПРИМЕНЕНИЕ ОБЪЕКТНО-ОРИЕНТИРОВАННОГО ПОДХОДА
ПРИ РАЗРАБОТКЕ АВТОМАТНОЙ МОДЕЛИ ПРОГРАММНЫХ СИСТЕМ 59

Доломанов А. А.
ДИСТАНЦИОННОЕ УПРАВЛЕНИЕ СРЕДСТВАМИ ИЗМЕРЕНИЙ
НА ПРИМЕРЕ SMB100 ROHDE&SCHAWZ 61

Калашников В. А., Чиркин К. Д.
РЕАЛИЗАЦИЯ СИСТЕМЫ ТРАНСФОРМАЦИИ БУЛЕВЫХ
ФОРМУЛ НА ЯЗЫКЕ PROLOG 63

Трусов Е. В.
QT QUICK CONTROLS 66

Родионов Д. А.
АВТОМАТИЗИРОВАННАЯ СИСТЕМА КОНТРОЛЯ РАБОТОСПОСОБНОСТИ
НА БАЗЕ СБИС 67

Шибанов С. В., Балашова И. Ю., Макарычев П. П.
ПЕРСПЕКТИВЫ И ПРОБЛЕМЫ ПРИМЕНЕНИЯ БЛОКЧЕЙНОВ ДЛЯ ПОСТРОЕНИЯ
РАСПРЕДЕЛЕННЫХ КОРПОРАТИВНЫХ СИСТЕМ 71

Шибанов С. В., Тарасеев А. В., Кикта Е. С.
КЛАССИФИКАЦИЯ ПРАВИЛ В АКТИВНЫХ БАЗАХ ДАННЫХ 74

Бородулин А. Р.
ЯЗЫК COLAMO И КОМПЛЕКС СРЕДСТВ ПРОГРАММИРОВАНИЯ
МНОГОКРИСТАЛЬНЫХ РЕКОНФИГУРИРУЕМЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ 76

Николаев А. В., Цыпин Б. В., Ярославцева Д. А.
ИСПОЛЬЗОВАНИЕ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ
ДЛЯ АНАЛИЗА СТРУКТУРНЫХ СХЕМ ИНФОРМАЦИОННО-ИЗМЕРИТЕЛЬНЫХ
И УПРАВЛЯЮЩИХ СИСТЕМ 80

Николаев А. В., Папко А. А., Поспелов А. В.
МЕТОДЫ ПОВЫШЕНИЯ ВИБРАЦИОННОЙ ИНТЕЛЛЕКТУАЛЬНОЙ
УСТОЙЧИВОСТИ 84

СЕТИ ЭВМ И ТЕЛЕКОММУНИКАЦИИ

Тарасов В. Н., Бахарева Н. Ф., Липилина Л. В.
МОДЕЛИРОВАНИЕ ТЕЛЕТРАФИКА В СЛУЧАЕ ШИРОКОГО
ДИАПАЗОНА ИЗМЕНЕНИЯ ЕГО ПАРАМЕТРОВ НА ОСНОВЕ
ТЕОРИИ МАССОВОГО ОБСЛУЖИВАНИЯ 88

Mustafa Sadeq Jaafar, Sergey A. Zinkin, Dmitry V. Pashchenko
RECONFIGURABLE NETWORK MODELS FOR DISTRIBUTED
COMPUTING SYSTEMS 92

Mustafa Sadeq Jaafar, Sergey A. Zinkin
THE IMPLEMENTATION OF GLOBAL COMPUTING THROUGH
THE MAPPING OF OBJECT-ORIENTED PETRI NETS INTO
THE ARCHITECTURE OF DISTRIBUTED COMPUTING SYSTEMS 105

Мартяшин Г. В., Пышкина И. С., Трокоз Д. А.
ОСОБЕННОСТИ ПОСТРОЕНИЯ КАРТ ПОКРЫТИЯ В СИСТЕМАХ ПАССИВНОГО МОНИТОРИНГА ТЕХНИЧЕСКИХ ПАРАМЕТРОВ СЕТЕЙ СОТОВОЙ СВЯЗИ 117

Мартяшин Г. В., Пышкина И. С.
АНАЛИТИЧЕСКИЙ ПРОГРАММНЫЙ ИНСТРУМЕНТ ПЛАНИРОВАНИЯ ЭКСПЕРИМЕНТА 120

Тумасов С. В., Гурин Е. И.
АНАЛИЗАТОР КАЧЕСТВА СВЯЗИ НА БАЗЕ ШУМОПОДОБНОГО СИГНАЛА 122

Никишин К. И.
АППАРАТНАЯ КОНФИГУРАЦИЯ ПЛИС ТИПА FPGA С ИСПОЛЬЗОВАНИЕМ ИНТЕРФЕЙСА ETHERNET ДЛЯ ОБМЕНА ИНФОРМАЦИЕЙ 124

Никишин К. И.
ОБМЕН ИНФОРМАЦИЕЙ С ИСПОЛЬЗОВАНИЕМ ИНТЕРФЕЙСА ETHERNET НА ОСНОВЕ ПЛИС ТИПА FPGA 128

Семенов А. О.
УСТРОЙСТВО, РЕАЛИЗУЮЩЕЕ АЛГОРИТМ ДИСПЕТЧЕРИЗАЦИИ НА БАЗЕ АДАПТИВНЫХ RAND-ПРИНЦИПОВ..... 130

Семенов А. О., Коннов Н. Н.
МОДЕЛИРОВАНИЕ АЛГОРИТМА СТОХАСТИЧЕСКОЙ ДИСПЕТЧЕРИЗАЦИИ ОЧЕРЕДЕЙ В ПАКЕТЕ..... 133

Аладьев Ю. Ю., Никишин К. И.
МОДЕЛИРОВАНИЕ СЕТЕВОГО ТРАФИКА В ПАКЕТЕ CPN TOOLS..... 135

Новиков М. Ю.
РАДИОСЕТЬ ПЕРЕДАЧИ ДАННЫХ SKYMAN/CA 139

Замятин Д. А., Богданова В. С., Степанова С. В.
КОНЦЕПЦИЯ НЕЗАВИСИМОЙ СПУТНИКОВОЙ КОМПЬЮТЕРНОЙ СЕТИ..... 142

Коннов М. Н.
ПОВЫШЕНИЕ ТОЧНОСТИ УСТАНОВКИ СЕТЕВОГО ВРЕМЕНИ ПРОТОКОЛОМ IEEE-1588 144

Кизилов Е. А., Коннов Н. Н., Патунин Д. В., Фролов И. Ю.
ОЦЕНКА РАЗМЕРОВ ОЧЕРЕДЕЙ В КОММУТАТОРЕ С ПОДДЕРЖКОЙ QOS..... 148

ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ И СЕТЕЙ

Егоров В. Ю.
ПЕРСПЕКТИВЫ ПРИМЕНЕНИЯ ОПЕРАЦИОННОЙ СИСТЕМЫ QP ОС В РАБОЧИХ СТАНЦИЯХ..... 153

Иванов А. И., Безяев А. В., Банных А. Г.
ХИ-КВАДРАТ ПРИБЛИЖЕНИЕ РАСПРЕДЕЛЕНИЯ ОШИБОК ВЫЧИСЛЕНИЯ ЭНТРОПИИ БИОМЕТРИЧЕСКОГО ОБРАЗА «ЧУЖОЙ» В ПРОСТРАНСТВЕ СВЕРТОК ХЭММИНГА 155

Волчихин В. И., Иванов А. И., Вятчанин С. Е.
ОСОБЕННОСТИ ОБУЧЕНИЯ СЕТЕЙ ВЕРОЯТНОСТНЫХ НЕЙРОНОВ «КРАМЕРА – ФОН МИЗЕСА» НА МАЛЫХ БИОМЕТРИЧЕСКИХ ВЫБОРКАХ 159

Иванов А. И., Малыгина Е. А.
ОСОБЕННОСТИ МНОГОМЕРНОЙ СТАТИСТИЧЕСКОЙ ОБРАБОТКИ БИОМЕТРИЧЕСКИХ ДАННЫХ..... 162

Фролов К. М., Дубравин А. В.
ГОРИЗОНТАЛЬНОЕ МАСШТАБИРОВАНИЕ ДАННЫХ В СОВРЕМЕННЫХ СУБД..... 163

Кормишина В. В.
ПРИМЕНЕНИЕ ЭЛЛИПТИЧЕСКИХ КРИВЫХ В КРИПТОГРАФИИ 167

Никишин К. И.
ШИФРОВАНИЕ ИНФОРМАЦИИ С ИСПОЛЬЗОВАНИЕМ
ПЕРЕСТАНОВОЧНОГО ШИФРА 171

Пащенко Д. В., Бальзанникова Е. А.
ПОВЫШЕНИЕ ТОЧНОСТИ ИДЕНТИФИКАЦИИ ПОЛЬЗОВАТЕЛЯ
ПО БИОМЕТРИЧЕСКИМ ДАННЫМ КЛАВИАТУРНОГО ПОЧЕРКА 173

Вашкевич Н. П., Мясин П. Ю., Поздняков С. Ю.
ПРИНЦИПЫ ПОСТРОЕНИЯ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ
В УСЛОВИЯХ СОВРЕМЕННОГО ИНФОРМАЦИОННОГО ПРОТИВОБОРСТВА 176

Яужев А. А.
КОМБИНИРОВАННОЕ СРЕДСТВО АУТЕНТИФИКАЦИИ 179

ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ И РАСПОЗНАВАНИЕ ОБРАЗОВ

Шушкевич Г. Ч., Шушкевич С. В.
ВИЗУАЛИЗАЦИЯ РЕШЕНИЙ ПРИКЛАДНЫХ ЗАДАЧ
С ИСПОЛЬЗОВАНИЕМ ОБЛАЧНЫХ ТЕХНОЛОГИЙ WOLFRAM CLOUD 182

Каперко А. Ф., Кулагин В. П.
ИНТЕЛЛЕКТУАЛЬНЫЕ МЕТОДЫ КОНТРОЛЯ ИОНИЗИРУЮЩЕГО
ИЗЛУЧЕНИЯ НА ОСНОВЕ СПЕКТРОМЕТРА С АЛМАЗНЫМИ ДЕТЕКТОРАМИ 185

Алкезуини М. М., Горбаченко В. И.
АДАПТАЦИЯ МЕТОДА ЛЕВЕНБЕРГА – МАРКВАРДТА
ДЛЯ ОБУЧЕНИЯ СЕТЕЙ РАДИАЛЬНЫХ БАЗИСНЫХ ФУНКЦИЙ 190

Бабич М. Ю.
ОПРЕДЕЛЕНИЕ СТЕПЕНИ ВОЗМОЖНОЙ ИНТЕЛЛЕКТУАЛИЗАЦИИ
ОРГАНИЗАЦИОННО-ТЕХНИЧЕСКИХ СИСТЕМ ПОСРЕДСТВОМ
ФУНКЦИИ ПРИНАДЛЕЖНОСТИ ЛИЦА, ПРИНИМАЮЩЕГО РЕШЕНИЯ,
К АВТОМАТИЗИРОВАННОЙ СИСТЕМЕ УПРАВЛЕНИЯ 194

Хоанг Тхай Хо
КОМПЛЕКС ПРОГРАММ ГЕОМЕТРИЧЕСКОГО МОДЕЛИРОВАНИЯ
ПРОСТРАНСТВЕННЫХ ФОРМ НА ОСНОВЕ ЧИСЛЕННОЙ
РЕАЛИЗАЦИИ ИНТЕРПОЛЯЦИОННЫХ АЛГОРИТМОВ 197

Токарев А. Н., Прошкин А. В.
МЕТОДИКА РАСЧЕТА ОБЪЕМОВ ПОДЗАДАЧ ПРИ РАВНОМЕРНОМ
ДЕЛЕНИИ В РАСПРЕДЕЛЕННЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМАХ
С НЕНАДЕЖНЫМИ УЗЛАМИ 202

Кузнецов Р. Н., Кузнецова О. Ю.
НЕЙРОСЕТЕВАЯ ДИАГНОСТИКА ПОСЛЕОПЕРАЦИОННЫХ
ОСЛОЖНЕНИЙ ПРИ ЖЕЛЧНОКАМЕННОЙ БОЛЕЗНИ 205

Абрамов И. А., Банникова А. Н., Мартышкина Н. А.
РЕАЛИЗАЦИЯ АЛГОРИТМА ПАРАЛЛЕЛЬНОГО ВЫДЕЛЕНИЯ
КОНТУРОВ ОБЪЕКТОВ НА ИЗОБРАЖЕНИИ 207

Алкезуини М. М.
СОВЕРШЕНСТВОВАНИЕ АЛГОРИТМОВ ГРАДИЕНТНОГО СПУСКА
ОБУЧЕНИЯ СЕТЕЙ РАДИАЛЬНЫХ БАЗИСНЫХ ФУНКЦИЙ 210

Валько А. Ф., Валько А. А.
СТРУКТУРА ОПТИЧЕСКИХ ИЛЛЮЗИЙ В НЕПОДВИЖНЫХ
ИЗОБРАЖЕНИЯХ ГРАФИЧЕСКИХ ОБЪЕКТОВ 215

Яремко О. Э.
РЕШЕНИЕ ПЛОСКИХ ЗАДАЧ ТЕОРИИ УПРУГОСТИ
СРЕДСТВАМИ MATLAB 216

Акинтеев М. А.
К ВОПРОСУ О МАТЕМАТИЧЕСКИХ МЕТОДАХ КОДИРОВКИ
ЦВЕТОВЫХ МОДЕЛЕЙ ПРИ ОБРАБОТКЕ ИЗОБРАЖЕНИЙ 219

Гладилин А. В.
КОНЦЕПЦИЯ БАЗЫ ЗНАНИЙ С ФУНКЦИЕЙ СЕМАНТИЧЕСКОГО
АНАЛИЗА ТЕКСТОВЫХ ДОКУМЕНТОВ 222

Гладилин А. В.
ОБЗОР СУЩЕСТВУЮЩИХ ПРОГРАММНЫХ КОМПЛЕКСОВ
ДЛЯ ПРОВЕДЕНИЯ КЛАСТЕРНОГО АНАЛИЗА 224

Савенков К. Е., Николаева В. А.
ОБУЧЕНИЕ СЕТЕЙ РАДИАЛЬНЫХ БАЗИСНЫХ ФУНКЦИЙ
ПРИ РЕШЕНИИ ЗАДАЧ АППРОКСИМАЦИИ 226

Никонов А. С., Бурукина И. П.
РАЗРАБОТКА КОМПОНЕНТОВ СИСТЕМЫ WAVELET-ПРЕОБРАЗОВАНИЯ
ИЗОБРАЖЕНИЯ 232

Горбачев Д. В.
МОДЕЛИРОВАНИЕ ПРОЦЕССА ИНСТРУМЕНТАЛЬНОЙ ДИАГНОСТИКИ
ЗАБОЛЕВАНИЯ СИСТЕМОЙ МАССОВОГО ОБСЛУЖИВАНИЯ
И ОТНОСИТЕЛЬНЫМИ ПРИОРИТЕТАМИ 236

Чернова Е. В., Полежаев П. Н.
АНАЛИЗ СУЩЕСТВУЮЩИХ ТЕХНОЛОГИЙ НЕЙРОННЫХ СЕТЕЙ 239

Мартышкин А. И.
РЕАЛИЗАЦИЯ ОПЫТНОГО ОБРАЗЦА РЕКОНФИГУРИРУЕМОЙ
ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЫ ДЛЯ ЦИФРОВОЙ ОБРАБОТКИ СИГНАЛА
НА БАЗЕ ПРОГРАММИРУЕМЫХ ЛОГИЧЕСКИХ ИНТЕГРАЛЬНЫХ СХЕМ 243

Романчук В. А.
ПОВЫШЕНИЕ ПРОИЗВОДИТЕЛЬНОСТИ НЕЙРОКОМПЬЮТЕРНЫХ
УСТРОЙСТВ ЗА СЧЕТ УПАКОВКИ ДАННЫХ 246

Осипов А. В.
АЛГОРИТМЫ, ПРИМЕНЯЕМЫЕ В ИГРАХ 250

Лукашенко В. В., Романчук В. А.
РАЗРАБОТКА АЛГОРИТМА СИНТАКСИЧЕСКОЙ ОБРАБОТКИ
НЕЙРОБАЗИСНОЙ ПРОГРАММЫ 252

Симаков А. А., Макурков А. С., Шибанов С. В.
СРАВНИТЕЛЬНЫЙ АНАЛИЗ ИНСТРУМЕНТОВ ВИЗУАЛИЗАЦИИ
ПОВЕДЕНИЯ ПРАВИЛ В АКТИВНЫХ БАЗАХ ДАННЫХ 255

Очередько О. О., Полежаев П. Н.
ВОЗМОЖНОСТИ ДЛЯ РАСПАРАЛЛЕЛИВАНИЯ НЕЙРОННЫХ СЕТЕЙ 258

Черепащук М. М.
РАЗРАБОТКА СИСТЕМЫ АВТОМАТИЧЕСКОЙ ГЕНЕРАЦИИ
ИГРОВЫХ СИТУАЦИЙ НА ПРИМЕРЕ RPG-ИГР 261

Жохов А. В.
НАПРАВЛЕНИЯ РАЗВИТИЯ СИСТЕМЫ ПРОГНОЗИРОВАНИЯ
И МОНИТОРИНГА ПОТРЕБНОСТЕЙ РЫНКА ТРУДА
В ПРОФЕССИОНАЛЬНЫХ КАДРАХ 264

Serikova Yu. I.
COMPARISON OF PREDICTORS WITH LINEAR COMPUTATIONAL
COMPLEXITY 266

Шполянская И. Ю., Загиров А. Р.
РАЗРАБОТКА WEB-ОРИЕНТИРОВАННОЙ РЕКОМЕНДАТЕЛЬНОЙ СИСТЕМЫ С ПРИМЕНЕНИЕМ ТЕХНОЛОГИЙ WEB MINING 269

Макарычев П. П.
ЭВРИСТИЧЕСКИЕ АЛГОРИТМЫ РЕШЕНИЯ ОПТИМИЗАЦИОННЫХ ЗАДАЧ В ОБЛАСТИ ИНФОРМАТИКИ 273

СИСТЕМЫ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ

Эпп В. В., Бурукина И. П.
АЛГОРИТМЫ РАЗМЕЩЕНИЯ И ТРАССИРОВКИ В МОБИЛЬНЫХ ИГРАХ 277

Войнов А. С., Сенокосов И. В., Чен-Вэй Янг
ТРАНСЛЯТОР SCL-ОПИСАНИЯ СЕТЕЙ SMART GRID В СИСТЕМЫ ФУНКЦИОНАЛЬНЫХ БЛОКОВ 280

Войнов А. С., Чен-Вэй Янг
ИСПОЛЬЗОВАНИЕ АЛГОРИТМА ЛИ В ТРАССИРОВКЕ СОЕДИНЕНИЙ В ДИАГРАММАХ SLD 283

Сенокосов И. В., Войнов А. С.
ПРОГРАММНАЯ МОДЕЛЬ СИСТЕМЫ СБОРКИ СЛОЖНЫХ ИЗДЕЛИЙ НА ОСНОВЕ МАГАЗИННЫХ АВТОМАТОВ-ПРЕОБРАЗОВАТЕЛЕЙ В СРЕДЕ NXTSTUDIO 286

Солопов Д. А., Мордасова Е. С., Калистратова И. В.
ПОДХОД К АВТОМАТИЗИРОВАННОМУ ВЫБОРУ ТИПА БЛОКА ШТАМПА 289

Семерич Ю. С., Тельнова Л. В.
ПРИМЕНЕНИЕ R-ФУНКЦИЙ ДЛЯ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ ГЕОМЕТРИЧЕСКИХ ОБЪЕКТОВ С ПЕРФОРАЦИЕЙ 290

Салмов Е. Н.
МОДЕЛИРОВАНИЕ УПРАВЛЯЕМЫХ ВЕНТИЛЬНО-ЭЛЕКТРОМЕХАНИЧЕСКИХ СИСТЕМ С ПЕРЕМЕННОЙ НАГРУЗКОЙ 294

Акопян С. А., Акопян В. Н., Мокрозуб В. А.
БАЗА ЗНАНИЙ СИСТЕМЫ АВТОМАТИЗИРОВАННОГО СОСТАВЛЕНИЯ ТЕХНОЛОГИИ ИЗГОТОВЛЕНИЯ ВАЛОВ 297

Бочаров В. А., Шивидова А. А.
РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ ДЛЯ СТРУКТУРИЗАЦИИ ТРЕБОВАНИЙ ДЛЯ ПРОЕКТИРОВАНИЯ КОНСТРУКТОРСКОЙ СТРУКТУРЫ ИЗДЕЛИЯ 301

Вилюха А. В.
АВТОМАТИЗАЦИЯ КОНТРОЛЯ КАЧЕСТВА ШТАМПОВАННЫХ КОНТАКТОВ С ПОМОЩЬЮ МЕТОДА СЖАТИЯ-РАСШИРЕНИЯ 305

Доброва Н. С.
АВТОМАТИЗИРОВАННЫЙ ВЫБОР ТЕХНОЛОГИЧЕСКИХ ПРОЦЕССОВ ТЕРМИЧЕСКОЙ ОБРАБОТКИ ИЗДЕЛИЙ ИЗ МЕТАЛЛОВ 309

Елагин М. С., Мокрозуб В. А., Храмцова Н. В.
ЗАДАЧА МИНИМИЗАЦИИ МЕТАЛЛОЕМКОСТИ ВЕРТИКАЛЬНЫХ АППАРАТОВ 311

Мананкова Е. В.
ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ РАБОТЫ ЭЛЕВАТОРНОГО КОМПЛЕКСА 313

Фролов И. Ю.
ОСОБЕННОСТИ ТРАССИРОВКИ ПЕЧАТНЫХ ПЛАТ И ПРИМЕНЕНИЕ БЛОКИРОВОЧНЫХ КОНДЕНСАТОРОВ 315

Шабалова К. Ю.
ШАБЛОН АРХИТЕКТУРЫ СИСТЕМЫ «МОДЕЛЬ–ПОВЕДЕНИЕ» ДЛЯ UNITY 3D 317

УПРАВЛЕНИЕ В СОЦИАЛЬНЫХ, ЭКОНОМИЧЕСКИХ И ТЕХНИЧЕСКИХ СИСТЕМАХ

Мкртчян В. С., Гамидуллаева Л. А.
МЕХАНИЗМЫ ЭФФЕКТИВНОГО ЦИФРОВОГО ВЗАИМОДЕЙСТВИЯ УЧАСТНИКОВ В ИННОВАЦИОННОЙ СИСТЕМЕ РЕГИОНА 322

Арестова М. В., Фионова Л. Р.
АНАЛИЗ ПРОЦЕССОВ ИСПОЛЬЗОВАНИЯ ЭЛЕКТРОННОЙ ПОДПИСИ НА ПРОМЫШЛЕННОМ ПРЕДПРИЯТИИ 324

Фионова Л. Р., Шибаева С. О.
РАЗРАБОТКА БАЗЫ ДАННЫХ ДЛЯ АВТОМАТИЗИРОВАННОГО СОЗДАНИЯ ДОЛЖНОСТНЫХ ИНСТРУКЦИЙ 327

Чечель П. К., Краснов А. А., Трокоз Д. А.
РАЗРАБОТКА КОНЦЕПТА ИНТЕРАКТИВНЫХ КАРТ КРИМИНАЛИЗАЦИИ РАЙОНОВ НА ОСНОВЕ ОФИЦИАЛЬНЫХ ДАННЫХ В ОТКРЫТЫХ ИСТОЧНИКАХ 330

Доброжанская П. С.
АВТОМАТИЗИРОВАННАЯ СИСТЕМА УПРАВЛЕНИЯ ЛОГИСТИКОЙ АВТОТРАНСПОРТНОГО ПРЕДПРИЯТИЯ 333

Курденкова Я. Г., Бурукина И. П.
РАЗРАБОТКА СИСТЕМЫ ПОДДЕРЖКИ ПРОВЕДЕНИЯ КОНФЕРЕНЦИИ 335

Лисенков А. В., Гюлзатян А. С., Поменкова Е. А.
ОСНОВНЫЕ ПРЕИМУЩЕСТВА ВЕДЕНИЯ ЭЛЕКТРОННОГО ФОРМУЛЯРА В ИНТЕРЕСАХ АВТОМАТИЗИРОВАННОГО УПРАВЛЕНИЯ ЖИЗНЕННЫМ ЦИКЛОМ ИЗДЕЛИЙ 337

Мазилкин Д. С.
СПОСОБЫ ОБРАБОТКИ WEB-СТРАНИЦ НА PHP 340

Давыдова Ю. В.
ПРОБЛЕМА ОБРАБОТКИ ОШИБОК В ТЕКСТАХ СООБЩЕНИЙ ПОЛЬЗОВАТЕЛЕЙ В ЗАДАЧЕ МОНИТОРИНГА ВИРТУАЛЬНЫХ СОЦИАЛЬНЫХ СЕТЕЙ 342

Джамбеков А. М.
ПРИНЯТИЕ ПАРЕТО-ОПТИМАЛЬНЫХ РЕШЕНИЙ ПРИ УПРАВЛЕНИИ ПРОЦЕССОМ КАТАЛИТИЧЕСКОГО РИФОРМИНГА 346

Люлякина Д. Н., Кузнецова О. Ю.
ОПТИМИЗАЦИЯ СИСТЕМЫ МЕНЕЖДМЕНТА ПРЕДПРИЯТИЯ: ПРОЦЕССНЫЙ ПОДХОД 348

Таланов Д. А.
РАЗРАБОТКА АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ УПРАВЛЕНИЯ ЖИЛЫМ ПОМЕЩЕНИЕМ 351

Тимонин А. Ю., Бождай А. С.
УРОВНИ МАТЕМАТИЧЕСКОГО ОПИСАНИЯ ПРОЦЕССА ПОСТРОЕНИЯ СОЦИАЛЬНОГО ПРОФИЛЯ 353

Тураев Р. А.
ПРИМЕНЕНИЕ WEB-ТЕХНОЛОГИЙ В РАЗРАБОТКЕ ИНФОРМАЦИОННЫХ СИСТЕМ 357

Баусова З. И., Старикова А. Ю., Шадрина Э. Ф.
АНАЛИЗ СТАТИСТИЧЕСКИХ ПОКАЗАТЕЛЕЙ ПАТЕНТНЫХ ДАННЫХ 360

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ОБРАЗОВАНИИ

Кревский И. Г., Антонов А. В.
ЭЛЕКТРОННАЯ ИНФОРМАЦИОННО-ОБРАЗОВАТЕЛЬНАЯ СРЕДА ВУЗА
КАК КЛЮЧЕВОЙ ЭЛЕМЕНТ ЕГО ИНФОРМАТИЗАЦИИ 365

Шмид А. В.
ЦИФРОВАЯ ЭКОНОМИКА – ВАШЕ БУДЕЩЕЕ. ЧТО ЭТО ТАКОЕ
И КАК НАЙТИ В НЕЙ СВОЕ МЕСТО 367

Волчихин В. И., Бершадский А. М., Бождай А. С.
СЕРВИС-ОРИЕНТИРОВАННАЯ АРХИТЕКТУРА СИСТЕМЫ УПРАВЛЕНИЯ
ЗНАНИЯМИ В ЭЛЕКТРОННЫХ ОБРАЗОВАТЕЛЬНЫХ СИСТЕМАХ 385

Глотова Т. В., Кревский И. Г.
ИСПОЛЬЗОВАНИЕ ВОЗМОЖНОСТЕЙ ЭЛЕКТРОННАЯ
ИНФОРМАЦИОННО-ОБРАЗОВАТЕЛЬНАЯ СРЕДА УНИВЕРСИТЕТА
В ПРЕПОДАВАНИИ IT ДИСЦИПЛИН 389

Баусова З. И., Прокофьев О. В., Шадрина Э. Ф.
АНАЛИЗ ПРОГРАММНЫХ ПРОДУКТОВ ЭКОНОМЕТРИЧЕСКОГО
МОДЕЛИРОВАНИЯ В УЧЕБНОМ ПРОЦЕССЕ 392

Ермолов И. А.
РАЗРАБОТКА WEB-ОРИЕНТИРОВАННОЙ ЭКСПЕРТНОЙ СИСТЕМЫ
ПОДДЕРЖКИ ТРУДОУСТРОЙСТВА ВЫПУСКНИКОВ ВУЗА
С ПРИМЕНЕНИЕМ ТЕХНОЛОГИЙ SEMANTIC WEB 395

Жук Д. М., Князева С. Ю., Маничев В. Б., Оглоблин Д. И.
ОБУЧЕНИЕ БАКАЛАВРОВ ОСНОВАМ СИСТЕМ
АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ В ТЕХНИЧЕСКОМ ВУЗЕ 398

Кочергин М. И.
КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ ПОЛЕТА ТЕЛА
В АТМОСФЕРЕ ДЛЯ ОБРАЗОВАТЕЛЬНЫХ ЦЕЛЕЙ 400

Сулипов Ш. Л., Шахмерзаева Х. Б., Ламаева Л. И.
ОРГАНИЗАЦИЯ ПРОЕКТНО-ИССЛЕДОВАТЕЛЬСКОЙ
ДЕЯТЕЛЬНОСТИ СТУДЕНТОВ НА УРОКАХ ИНФОРМАТИКИ 404

Чечель П. К., Краснов А. А., Пащенко Д. В.
СРАВНИТЕЛЬНЫЙ АНАЛИЗ ОРГАНИЗАЦИИ УЧЕБНОГО ПРОЦЕССА
В ВЫСШИХ УЧЕБНЫХ ЗАВЕДЕНИЯХ РОССИИ И ЕВРОПЫ.
РАЗНИЦА МЕТОДОВ И ТЕХНОЛОГИЙ 406

Брусникин Е. С., Кичук В. И., Коннов Н. Н.
ГРАФИЧЕСКИЙ РЕДАКТОР ДИАГРАММ МИКРОПРОГРАММНОЙ ЛОГИКИ 409

Научное издание

НОВЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И СИСТЕМЫ

СБОРНИК НАУЧНЫХ СТАТЕЙ
XIV Международной научно-технической
конференции,
посвященной 70-летию кафедры «Вычислительная техника»
и 30-летию кафедры «Системы автоматизированного проектирования»

г. Пенза, 22–24 ноября 2017 г.

Материалы печатаются в авторской редакции.

Компьютерная верстка *Р. Б. Бердниковой*
Дизайн обложки *А. А. Стаценко*

Подписано в печать 20.11.2017.
Формат 60×84⅛. Усл. печ. л. 48,83.
Тираж 70. Заказ № 695.

Издательство ПГУ.
440026, Пенза, Красная, 40.
Тел./факс: (8412) 56-47-33; e-mail: iic@pnzgu.ru