

На правах рукописи

ЗИНКИН Сергей Александрович

**РАЗВИТИЕ ТЕОРЕТИЧЕСКИХ ОСНОВ И МЕТОДОВ
ФУНКЦИОНАЛЬНО-СТРУКТУРНОЙ ОРГАНИЗАЦИИ СИСТЕМ
И СЕТЕЙ ВНЕШНЕГО ХРАНЕНИЯ И ОБРАБОТКИ ДАННЫХ**

Специальность 05.13.15 – Вычислительные машины и системы

**Специальность 05.13.13 – Телекоммуникационные системы
и компьютерные сети**

А в т о р е ф е р а т
диссертации на соискание ученой степени
доктора технических наук

ПЕНЗА 2009

Работа выполнена в государственном образовательном учреждении высшего профессионального образования «Пензенский государственный университет».

Научный консультант – доктор технических наук, профессор,
заслуженный деятель науки и техники
Российской Федерации
Вашкевич Николай Петрович.

Официальные оппоненты: доктор технических наук, профессор
Огнев Иван Васильевич;
доктор технических наук, профессор
Борисов Вадим Владимирович;
доктор физико-математических наук,
профессор
Голованов Олег Александрович.

Ведущее предприятие – ФГУП "Пензенский научно-исследовательский электротехнический институт".

Защита состоится 24 декабря 2009 г., в 14 часов, на заседании диссертационного совета Д 212.186.01 в государственном образовательном учреждении высшего профессионального образования «Пензенский государственный университет» по адресу: 440026, г. Пенза, ул. Красная, 40.

С диссертацией и авторефератом можно ознакомиться в библиотеке государственного образовательного учреждения высшего профессионального образования «Пензенский государственный университет». Автореферат размещен на сайте ВАК.

Автореферат разослан «___» _____ 2009 г.

Ученый секретарь
диссертационного совета
доктор технических наук,
профессор

Гурин Е. И.

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность темы. Современные системы хранения и обработки данных содержат многие компоненты, созданные разными производителями, – серверы, коммуникационное оборудование, накопители информации, файловые и операционные системы, системы управления базами данных. В данных системах реализуются сложные взаимодействия между указанными компонентами и с внешней средой. Современные хранилища данных могут насчитывать тысячи процессоров, дисковых накопителей информации и могут хранить и обрабатывать петабайты данных. Под системами или сетями внешнего хранения и обработки данных в диссертационной работе подразумеваются такие системы или сети обработки данных, основой построения и развития которых являются хранилища, подключенные к сети, сети хранения данных или сети интеллектуальных многофункциональных ячеек хранения данных.

Организация вычислений и хранение пользовательских данных в вычислительных сетях являются перспективными и быстроразвивающимися областями как в промышленности, так и в научных исследованиях. В этой связи актуальным является рассмотрение всех основных аспектов архитектурной организации и использования аппаратных и алгоритмических средств сетей и систем хранения и обработки данных. Технология хранения данных рассматривается как развитие технологий открытых систем, способствующее, в свою очередь, развитию файловых вычислений с общим доступом через сеть к хранимой информации. К наиболее актуальным разработкам в области сетевого хранения данных относятся также интеллектуальные процессоры, взаимодействующие с традиционными подсистемами хранения данных и обеспечивающие управление массивами хранения данных, кэшированием команд и данных, а также управление взаимодействиями по каналам связи. В совокупности указанные средства должны обеспечивать множественный доступ к данным, масштабируемость, виртуализацию, внешнюю и внутреннюю интеллектуализацию хранения и обработки данных.

Решению проблем создания систем и сетей хранения и обработки данных посвящены работы многих исследователей в нашей стране и за рубежом. Однако в существующих работах недостаточное внимание уделено описанию реализаций систем и сетей, построение которых основано на интеграции и консолидации ресурсов внешнего хранения с ресурсами удаленной обработки данных, на концепциях структурирования декларативных и процедурных знаний при архитектурном проектировании, что определяет **основную проблемную ситуацию**. Известные разработки основаны на практическом опыте разработчиков, применении содержательных и неформализованных концептуальных моделей, что затрудняет проектирование и увеличивает сроки реализации проекта.

В подсистемах управления хранением и обработкой данных целесообразно использовать элементы систем баз данных и знаний, интегрированные с подсистемами кэширования. Узлы, или точки виртуализации развитых интеллектуальных систем и сетей хранения данных, помимо функций управления доступом и представления данных, управления метаданными и структурами данных, должны выполнять функции объектно-ориентированного хранения данных, в том числе структурированных. При работе они могут использовать собственную или разделяемую базу данных с информацией для управления. На данной основе целесообразно строить сетевые самоуправляемые интеллектуальные хранилища данных, содержащие в своем составе менеджеры работы с файлами, функции которых дополнены функциями для работы с отношениями реляционной базы данных, а также независимые менеджеры, обладающие доступом к общим данным через сеть, возможностью файловых вычислений, обеспечивая эффективное выполнение сетевых приложений.

С точки зрения конечных пользователей сеть внешнего хранения и обработки данных целесообразно представить в виде внешнего распределенного виртуального сопроцессора с персональными областями хранения и возможным совместным использованием данных. В этой связи особенно важно появление провайдеров сетевых услуг по хранению данных, обеспечивающих пользователей ресурсами хранения. Сеть при этом должна служить инфраструктурой доступа к глобальной системе хранения данных. Целесообразно предоставлять аналогичные услуги и по обеспечению пользователей вычислительными ресурсами через сеть.

Актуальность научных исследований в области внутренней и внешней интеллектуализации, архитектурного моделирования и проектирования систем и сетей внешнего хранения и обработки данных на основе интеграции формальных представлений определяется необходимостью создания новых эффективных методов анализа и синтеза процессов управления информационным обменом, хранением и обработкой информации в системах и сетях следующих поколений. Актуально развитие теоретических основ функционально-структурной организации систем и сетей внешнего хранения обработки данных, задачи повышения функциональных возможностей и производительности в которых решаются за счет декларативно-процедурной структуризации знаний о функционировании систем.

Целью работы является теоретическое обоснование и разработка методов архитектурного моделирования и функционально-структурной организации систем и сетей внешнего хранения и обработки данных, обладающих повышенной производительностью и расширенными функциональными возможностями.

Для достижения поставленной цели научного исследования в диссертационной работе решены следующие основные задачи:

– проведен анализ современного состояния и тенденций развития систем и сетей внешнего хранения и обработки данных применительно к задачам построения проблемно-ориентированных систем, на основании чего даны предложения по уменьшению последствий семантического разрыва между инфраструктурой систем хранения данных и алгоритмическими структурами, обеспечивающими многофункциональность системы;

– разработаны и теоретически обоснованы методы архитектурной и функционально-структурной организации систем и сетей внешнего хранения и обработки данных, позволившие сократить затраты на реализацию систем и сетей внешнего хранения и обработки данных и повысить качество принимаемых решений;

– исследованы и развиты формальные модели, обеспечившие поддержку архитектурного моделирования и функционально-структурной организации систем и сетей внешнего хранения и обработки данных;

– исследованы и реализованы в технологии “активной классной доски” методы внешней и внутренней интеллектуализации систем и сетей внешнего хранения и обработки данных;

– разработан метод решения задачи сквозного архитектурного и функционально-структурного проектирования многофункциональных распределенных систем управления внешней памятью ЭВМ и систем, основанный на развитии концептуальных моделей до определенных формально имитационных поведенческих моделей и формальных непосредственно исполняемых спецификаций;

– обоснован и исследован синтез структур и алгоритмов функционирования систем и сетей хранения и обработки данных на базе виртуализации и интеграции сетевых ресурсов хранения данных с вычислительными ресурсами;

– разработаны, обоснованы и исследованы концептуальные и формально определенные имитационные поведенческие модели, адекватная данным моделям сетевая интерпретирующая среда и реализованы системы синтеза структур и программного обеспечения систем и сетей внешнего хранения и обработки данных;

– предложены архитектурные решения для основных классов систем и сетей хранения и обработки данных, обладающих повышенной производительностью и развитыми функциональными возможностями.

Объектом исследования являются системы и сети внешнего хранения и обработки данных с расширенными функциональными возможностями по обеспечению управления трафиком передаваемых данных, хранения и обработки структурированной и неструктурированной информации.

Предметом исследования являются методы управления использованием ресурсов, информационным обменом, хранением и обработкой данных, функционально-структурная организация систем и сетей внешнего хране-

ния и обработки данных, алгоритмическое и методическое обеспечение технологий архитектурного моделирования и функционально-структурного проектирования.

Методы исследования. В диссертационной работе используются теоретико-множественные и логико-алгебраические представления, модели и методы искусственного интеллекта, методы построения поведенческих имитационных моделей, функционально-структурный подход к анализу и проектированию, концепции интеграции и дифференциации функций и структур систем.

Научная новизна заключается в развитии теоретических основ архитектурной, функционально-структурной организации и виртуализации систем и сетей внешнего хранения и обработки данных, в том числе:

- в теоретическом обосновании методов построения концептуальных и формализованных имитационных поведенческих моделей сложных дискретно-событийных систем, отличающихся тем, что на основе применения концептуальных графов, сценарных представлений и абстракций сетей виртуальных машин обеспечивается эффективное решение задач сквозного архитектурного и функционально-структурного проектирования систем и сетей внешнего хранения и обработки данных;

- в развитии формальных моделей, обеспечивающих поддержку архитектурного моделирования и анализ функционально-структурной организации систем и сетей внешнего хранения и обработки данных, отличающихся более адекватным представлением аппаратных и программных средств;

- в теоретическом обосновании систем согласования и координации процессов и объектов как базовой платформы для построения прототипного и рабочего аппаратно-программного обеспечения систем и сетей внешнего хранения и обработки данных в отличие от известных, основанном на применении предложенных в диссертационной работе эволюционирующих иерархических алгебраических систем, иерархических сетей абстрактных машин, что обеспечивает высокий уровень адекватности формальных представлений содержательному описанию;

- в развитии известных и в разработке новых методов внешней и внутренней интеллектуализации управления в системах и сетях внешнего хранения данных, позволяющих расширить границы применимости методов и моделей искусственного интеллекта для данной предметной области;

- в развитии принципов архитектуры, отличной от фон-неймановской на системном уровне и способствующей развитию принципов прозрачности в управлении ресурсами и масштабируемости, организации эффективного множественного доступа к средствам внешнего хранения и обработки данных;

- в формализованном описании архитектуры, топологии и функционирования систем и сетей внешнего хранения и обработки данных, отличающихся использованием алгебраического подхода к синтезу иерархических сценарных моделей функционирования, иерархических сетей абстрактных

машин и пространственных сценариев развертывания ресурсов хранения и обработки данных; данный подход позволяет сократить затраты на реализацию систем и сетей внешнего хранения и обработки данных и повысить качество принимаемых решений, в том числе за счет инвариантности разрабатываемого алгоритмического обеспечения относительно платформы реализации;

- в выявлении и реализации типичных структурных конфигураций сценариев, в результате чего осуществлена интеграция технологии “классной доски” с мультиагентной технологией, причем “классная доска” становится активным объектом, а решатели-мультиагенты, используя ее содержимое (возможно, по инициативе самой “классной доски”), реализуют распределенное выполнение, взаимодействия и синхронизацию реализуемых процессов.

Практическая ценность работы связана с архитектурными, функционально-структурными решениями и с разработкой средств проектирования и реализации систем и сетей внешнего хранения и обработки данных:

- предложены методы практической реализации уровней параллельности в сложных системах внешних запоминающих устройств (ВЗУ) и разработаны способы распараллеливания передачи данных;

- предложены методы реализации сетевых архитектур хранения данных с блочной и файловой виртуализацией;

- предложены способы реализации архитектуры узлов хранения данных в глобальных сетях хранения на основе концептуальной схемы использования многофункционального сетевого ВЗУ;

- исследованы и реализованы методы и технологии проектирования средств обработки структурированной информации, алгоритмического и программного обеспечения в сетях хранения данных;

- предложена функционально-структурная организация многофункционального узла сети внешних запоминающих устройств, включающего кэш-память;

- предложена функционально-структурная организация многофункциональной системы внешних запоминающих устройств, подключаемой к вычислительной сети;

- предложена функционально-структурная организация системы внешних запоминающих устройств, включающей машину баз данных;

- предложена функционально-структурная организация на основе концепции согласования процессов и объектов и реализация в сетевой среде процессора базы данных;

- разработаны способы пространственно-временной организации управляющих воздействий в подсистемах согласования и координации процессов и объектов систем и сетей внешнего хранения и обработки данных, обеспечивающие эффективные взаимодействия компонент;

- предложена и реализована концепция построения сетей хранения и обработки данных, предназначенных для выполнения сопроцессорных

функций хранения и обработки данных клиентских приложений, базирующаяся на технологии активной “классной доски” и обеспечивающая внутреннюю и внешнюю интеллектуализацию данных систем;

- реализованы в реальной сетевой среде основные уровни абстракций сети, используемой для виртуализации и моделирования сложных многофункциональных систем и сетей хранения и обработки данных;

- предложены и реализованы принципы структуризации разнообразных в семантическом и синтаксическом отношениях системных и процедурных знаний о функционально-структурной организации систем и сетей внешнего хранения и обработки данных;

- созданы система синтеза структур и программного обеспечения и система спецификации и верификации систем и сетей внешнего хранения и обработки данных.

Основные результаты, выносимые на защиту:

1. Метод архитектурного моделирования и проектирования проблемно-ориентированных систем и сетей внешнего хранения и обработки данных, обеспечивающий согласованность исходной концептуальной модели с формально определенными в процессе иерархического проектирования имитационными поведенческими моделями.

2. Теоретическое обоснование на основе определенных в работе эволюционирующих иерархических алгебраических систем, иерархических сетей абстрактных машин основного класса систем согласования и координации процессов и объектов как базовой платформы для построения прототипного и рабочего программного и аппаратного обеспечения систем и сетей внешнего хранения и обработки данных, обеспечивающей более адекватное отображение сети алгоритмических модулей на физическую сеть при реализации структурных способов повышения производительности.

3. Метод “активной классной доски” и его использование в качестве основы технологии внешней и внутренней интеллектуализации систем и сетей внешнего хранения и обработки данных. Эффективность реализации данного метода обеспечивается внедрением механизмов, основанных на формальных исполняемых спецификациях и на декларативно-процедурной структуризации представления знаний о предметной области, что сокращает затраты на реализацию систем внешнего хранения и обработки данных в аппаратно-программных средах, обеспечивающих согласование и координацию процессов и объектов.

4. Концепция развития архитектуры, отличной от фон-неймановской на системном уровне реализации систем и сетей хранения и обработки данных и способствующей развитию принципов прозрачности в управлении ресурсами и масштабируемости, организации множественного доступа к средствам внешнего хранения и обработки данных и облегчающей интеграцию ресурсов хранения с вычислительными ресурсами.

5. Принципы функционально-структурной организации на основе формального описания концепции согласования и координации процессов и объектов. Использование указанных принципов позволяет сократить затраты на реализацию основных методов управления в функционально-централизованных, функционально-децентрализованных и топологически-распределенных системах и сетях хранения и обработки данных за счет адекватности формальных спецификаций результирующей сети алгоритмических модулей.

6. Метод интерпретации пространственных и временных сценариев функционирования систем и сетей хранения и обработки данных сетями абстрактных машин, предоставляющий возможность эволюционного перехода от концептуальных моделей к непосредственно исполняемым формальным логико-алгебраическим спецификациям.

7. Метод расширения функциональных возможностей систем и сетей внешнего хранения и обработки данных за счет включения дополнительных функций обработки данных в узлах (точках) виртуализации, позволяющий реализовать логическую архитектуру на основе произвольной архитектуры вычислительных средств.

8. Концептуальные, имитационные поведенческие модели систем и сетей хранения и обработки данных, сетевая среда для реализации получаемых формальных спецификаций, структура и реализация системы синтеза структур и алгоритмического обеспечения систем и сетей внешнего хранения и обработки данных, позволяющие осуществить архитектурную и функционально-структурную организацию данных систем и сетей с развитыми функциональными возможностями и высокой производительностью.

9. Архитектурная и функционально-структурная организация: многофункционального узла сети внешних запоминающих устройств, включающего кэш-память; многофункциональной системы внешних запоминающих устройств, подключаемой к вычислительной сети; системы внешних запоминающих устройств, включающей машину баз данных; процессора базы данных, реализуемого в сетевой среде; сетевой среды для реализации виртуализированных архитектур систем и сетей внешнего хранения и обработки данных. Эффективность функционально-структурной организации данных систем и сетей при ее реализации обусловлена ее унифицированным логико-алгебраическим описанием на всех уровнях моделирования и проектирования, что снижает затраты на структурную интерпретацию данных описаний в программно-аппаратных средах.

Реализация и внедрение результатов диссертационной работы.

Диссертационная работа выполнялась в рамках научно-исследовательских работ, проводимых на кафедре "Вычислительная техника" Пензенского государственного университета в области разработки архитектур, методов и средств проектирования систем и сетей хранения и обработки данных, в том числе в рамках НИР, финансируемых Министерством образования

и науки РФ: “Теория и методы организации управления распределенными вычислительными процессами в многопроцессорных вычислительных системах и метакомпьютерных сетях”, № гос. регистр. 01.2.00 315092 (2003–2004 гг.) и “Теория, методы и средства организации систем параллельной обработки информации на основе использования недетерминированных автоматов”, № гос. регистр. 0120.0 502707 (2005–2009 гг.). Разработанные в ходе диссертационной работы системы включены в Государственный фонд алгоритмов и программ, в том числе “Система имитационного моделирования многоуровневых функционально-децентрализованных вычислительных систем” (Инв. № 50870001130), “Разработка и исследование моделей подсистем дисковой памяти с буферной памятью” (Инв. № 50870000284) и “Разработка и исследование перспективных систем ВЗУ на магнитных дисках” (Инв. № 50890001105).

Основные положения диссертационной работы внедрены на промышленных предприятиях ОАО «НПП “Рубин”», НИКИРЭТ – филиал ФГУП ФНПЦ и ПО “Старт им. М. В. Проценко”; использование результатов подтверждено соответствующими актами о внедрении.

Научные и практические результаты работы включены в ряд курсов лекций на кафедре “Вычислительная техника” Пензенского государственного университета, использованы в лабораторных практикумах, курсовом и дипломном проектировании, а также в НИР студентов. Результаты работы использованы при подготовке 12 учебных и учебно-методических пособий.

Апробация работы. Основные научные положения и результаты диссертационной работы докладывались и обсуждались на научно-практических конференциях и семинарах; при этом автором сделано 52 доклада на международных конференциях, 22 доклада на всесоюзных и всероссийских и 7 докладов на региональных конференциях и семинарах в Москве, Санкт-Петербурге, Киеве, Новосибирске, Минске, Казани, Воронеже, Саратове, Твери, Рязани, Харькове, Челябинске, Пензе, Тамбове, Курске, Варне (Болгария), Костенеце (Болгария) и в других городах. Результаты работы также регулярно докладывались на кафедре “Вычислительная техника” Пензенского государственного университета.

Публикации. По теме диссертации опубликовано 82 печатные работы, в том числе 40 статей (15 статей опубликовано в изданиях, рекомендованных ВАК РФ), 42 публикации – в материалах докладов на международных и региональных конференциях и авторское свидетельство. Все результаты, составляющие содержание диссертации, получены автором самостоятельно.

Структура и объем диссертационной работы. Работа состоит из введения, шести глав основной части, заключения, изложенных на 479 страницах (включая 143 рисунка и 4 таблицы), шести приложений на 48 страницах, (включая 18 рисунков и 3 таблицы) и списка литературы из 248 наименований.

КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

Во введении обоснована актуальность темы, сформулированы цель и задачи диссертационной работы, определены направления исследования, научная новизна и практическая значимость работы.

В первой главе проанализировано построение высокопроизводительных многофункциональных систем и сетей хранения и обработки данных, выделены основные особенности системного моделирования и архитектурного проектирования данных систем, даны рекомендации по выбору перспективных сетевых платформ и методов функционально-структурного проектирования; приведены примеры реализации систем и сетей внешнего хранения и обработки данных, базирующиеся на разработках автора.

При создании распределенных вычислительных систем особое значение придается разработке высокопроизводительных, надежных и интеллектуальных систем внешней памяти ЭВМ. Одним из перспективных путей решения данной задачи является использование сетевых архитектур ВЗУ: упрощается организация коллективного доступа к ВЗУ и реконфигурация системы, возможно включение разнообразных сетевых служб и серверов, увеличивающих уровень “интеллектуальности” системы ВЗУ, повышается производительность системы за счет параллельного доступа к ВЗУ в сети. Системы ВЗУ, построенные на основе сетевых архитектур, принято называть сетями ВЗУ. Используемое в диссертационной работе понятие “внешнее хранение и обработка данных” подчеркивает тот факт, что рассматриваются такие системы и сети обработки данных, ядром которых является система или сеть хранения данных.

Современные тенденции развития систем внешней и массовой памяти подтверждают перспективность и целесообразность использования сетевой идеологии при их создании: становится возможным объединение систем ВЗУ, повышается надежность хранения данных, становятся разнообразными услуги хранения данных, эффективнее используются все системы хранения данных, унифицируется доступ к различным устройствам хранения независимо от их размещения, упрощается масштабирование и консолидация ресурсов хранения, снижаются административные расходы при эксплуатации. Однако традиционные системы хранения данных зачастую не в полном объеме удовлетворяют современные потребности пользователей. Современные системы хранения должны обладать возможностью организации взаимодополняющих уровней хранения данных, а также обеспечивать масштабируемую виртуализацию и интеграцию вычислительных ресурсов и ресурсов хранения. Интеллектуальные дисковые массивы и узлы виртуализации при определенных условиях могут быть также основой и для организации пула вычислительных ресурсов, обеспечивая при этом реализацию согласованных распределенных вычислений. Само хранение данных можно

рассматривать как приложение, использующее файловую или блочную виртуализацию.

Определение архитектуры систем хранения и обработки данных обычно включает описание признаков и свойств структурной и функциональной организации. Расширенное определение архитектуры дополняется характеристикой логической, семантической и информационной организации систем и перечислением компонент. В процессе функционирования некоторой системы ее структурная, функциональная, логическая и семантическая организации могут модифицироваться в зависимости от внешних условий. Понятия и определения, характеризующие архитектуру системы хранения и обработки данных, в общем случае соответствуют разнообразным абстракциям. Важнейшей среди них является абстрактная модель функциональной организации системы, определяющая взаимодействия всех ее компонент. Особую роль понятие абстракции играет в определении иерархических виртуальных систем, формальное описание спецификаций для которых развито недостаточно. Методы формализации должны охватывать и преобразования виртуальных машин с учетом их иерархической структуризации.

Принятая концепция виртуализации способна обеспечить также консолидированное логическое представление узлов системы внешнего хранения и обработки данных. Особое значение для реализации принятой в диссертационной работе концепции “внешнего” хранения и обработки данных имеет реализация централизованного (с точки зрения одного пользователя) управления вычислениями и данными. Для каждого пользователя в данном случае может быть задан виртуальный менеджер, выполняющий заданный сценарий работы.

Определенные в диссертационной работе принципы проектирования систем и сетей внешнего хранения и обработки данных позволяют построить модели, адекватные проектируемым системам, и последовательно использовать их на этапах проектирования, сопровождаемых содержательным, концептуальным и имитационным поведенческим моделированием. Основные проблемы системного и функционально-структурного проектирования сетей и систем ВЗУ связаны также с использованием “интеллектуальных” контроллеров и коммутаторов. Ключевое слово здесь взято в кавычки, поскольку обычно в литературе под ним не подразумевается использование принципов систем искусственного интеллекта. В отличие от указанной трактовки при рассмотрении функционально-структурной организации системы или сети хранения и обработки данных с позиций искусственного интеллекта, предложено считать, что в некоторой базе знаний хранятся формализованные декларативные и процедурные знания об архитектуре и функционировании системы. Декларативные знания связаны с пространственными, количественными и, возможно, классификационными отношениями объектов. Процедурные знания указывают на последовательность

действий, которая должна быть выполнена в системе с учетом сложившейся ситуации при обработке данных. Подобные действия приводят также к пополнению представления текущей ситуации новыми фактами. Для представления декларативных знаний о предметной области в диссертационной работе предложено использовать семантические сети. Декларативные и процедурные знания о структуре и функционировании систем и сетей хранения и обработки данных предложено задавать с помощью сценарных сетей – разновидностей семантических сетей с событиями и темпоральными отношениями между ними.

Построение новых архитектур, концептуальных, математических и алгоритмических моделей, изложенных в последующих главах, проведено с учетом опыта авторских разработок, описанных в первой главе: организация параллельной работы накопителей на магнитных дисках на нижнем уровне иерархии системы ВЗУ, реализация сети ВЗУ с виртуализацией на уровне блоков данных, реализация узла сетевой внешней памяти с виртуализацией на уровне файлов и блоков.

Вторая глава посвящена разработке, развитию и интеграции формализмов, предназначенных для поддержки новых архитектурных направлений, в том числе виртуализации, внутренней и внешней интеллектуализации систем и сетей внешнего хранения данных.

Для унификации описаний систем рассматриваемого класса на различных уровнях абстракций здесь используются логико-алгебраические представления, в связи с чем решаются задачи совершенствования логико-алгебраического аппарата описания дискретных систем с событийно-временной координацией взаимодействующих процессов и в дальнейшем разработки на этой основе формальных средств спецификации имитационных моделей.

Предлагаемые средства спецификации допускают простую программную реализацию имитационных моделей и, после несложных преобразований, обеспечивают реализацию системы алгоритмических модулей в виде совокупности управляющих программ вычислительной системы или сети. Данные преобразования обеспечивают переход от временной координации модулей имитационных моделей к причинно обусловленной координации модулей управляющих программ. Подобные спецификации, используемые на различных этапах архитектурного и функционально-структурного проектирования систем и сетей хранения и обработки данных, названы непосредственно интерпретируемыми формальными спецификациями. Особенностью используемых в этих целях логико-алгебраических описаний является возможность их применения на ранних этапах иерархического проектирования при переходе от содержательных описаний к концептуальным моделям и далее к определяемым формально имитационным поведенческим моделям. На конечном этапе проектирования формируется и реализуется в

сетевой среде система алгоритмических модулей, полностью адекватная исходной модели.

Для концептуального представления принципов организации распределенных систем с целью выработки единых в концептуальном плане подходов предлагается использовать формализм сетей абстрактных машин, берущих свое начало от машин Колмогорова–Успенского–Шёнхаге и Гуревича. При определении декларативной составляющей формализма используется понятие алгебраических систем, а для представления процедурной составляющей в сетях абстрактных машин и в сценарных сетях, определенных в дальнейшем, используются некоторые элементы алгебры алгоритмов Глушкова, в частности, операции α -дизъюнкции и α -итерации для всюду определенных условий и понятие суперпозиции операций.

Сеть абстрактных машин (СеАМ) определяется следующим набором:

$$N = (A_1, \dots, A_n, \Sigma, I(t), I(t_0), \Sigma', J, \mu_a, U_F, U_P, M, \Omega_{Top}, \Omega_{Cop}, L, \Omega_{Lop}, \Phi, B, \Omega_B, \mu_{update}, \mu_{test}, \mu_{block}, \mu_B, \mu_{MB}, \mu_L, C, Q, W), \quad (1)$$

где $A_1, \dots, A_n$ – непустые непересекающиеся множества (основы), на которых определены множества функций и предикатов, имена которых составляют “текущую” (с изменяющейся, или эволюционирующей в процессе функционирования сети интерпретацией $I(t)$ функциональных и предикатных символов) и “статическую” (с неизменяющейся интерпретацией) сигнатуры;

$A = \{A_1, \dots, A_n\}$ – множество основ;

$\Sigma = F \cup P$ – каузально-объектная сигнатура ($F = F_C \cup F_{Ob}$, $P = P_C \cup P_{Ob}$);

F_C – множество “каузальных” функций;

F_{Ob} – множество “объектных” функций;

P_C – множество “каузальных” предикатов;

P_{Ob} – множество “объектных” предикатов;

$I(t)$, $t \geq t_0$, – текущая интерпретация функциональных и предикатных символов, составляющих сигнатуру Σ ;

$I(t_0)$, $t_0 \geq 0$, – начальная интерпретация функциональных и предикатных символов, составляющих сигнатуру Σ ;

$\Sigma' = F' \cup P'$ – статическая сигнатура (с не изменяющейся в процессе функционирования СеАМ интерпретацией функциональных символов из множества F' и предикатных символов из множества P');

J – интерпретация функциональных символов из множества F' и предикатных символов из множества P' ;

$\mu_a : (\Sigma \cup \Sigma') \rightarrow \{1, 2, \dots, n_a\}$ – отображение аргументности, где $n_a = \text{const}$; в процессе функционирования СеАМ аргументности функций и предикатов остаются неизменными;

U_F – множество элементарных обновлений функций, сопоставленных функциональным символам из Σ ;

U_P – множество элементарных обновлений предикатов, сопоставленных предикатным символам из Σ ;

$U = U_F \cup U_P$ – система образующих алгебры модулей;

M – множество модулей СеАМ, реализующих локальные преобразования (обновления) текущей интерпретации $I(t)$ сигнатуры Σ ; под модулем для простоты подразумевается СеАМ-выражение, не содержащее свободных вхождений предметных переменных;

Ω_{Top} – система темпоральных операций вида $\omega_{Top} : M \times M \rightarrow M$, принимающих значения в множестве модулей M ;

Ω_{Cop} – система дополнительных операций вида $\omega_{Cop} : L \times M \times M \rightarrow M$ и $\omega'_{Cop} : L \times M \rightarrow M$ (“управляющих конструкций” СеАМ), принимающих значения в множестве модулей M ;

L – множество логических условий;

Ω_{Lop} – система логических операций вида $\omega_{Lop} : L \times L \rightarrow L$ и $\omega'_{Lop} : L \rightarrow L$, принимающих значения в множестве логических условий L ;

Φ – множество атомарных формул (атомов) – система образующих алгебры логических условий;

B – множество блоков СеАМ;

Ω_B – множество операций, выполняемых в блоках модулей СеАМ;

$\mu_{update} : M \rightarrow \mathbf{P}(\Sigma) \setminus \emptyset$ – отображение, сопоставляющее каждому модулю подмножество функций и предикатов, которые им модифицируются ($\mathbf{P}$ – символ булеана):

$\mu_{test} : M \rightarrow \mathbf{P}(\Sigma) \setminus \emptyset$ – отображение, сопоставляющее каждому модулю подмножество функций и предикатов, которые им проверяются;

$\mu_{block} : M \rightarrow \mathbf{P}(\Sigma) \setminus \emptyset$ – отображение, сопоставляющее каждому модулю подмножество функций и предикатов, с которыми оперирует данный модуль, причем

$$(\forall m \in M)(\mu_{block}(m) = \mu_{update}(m) \cup \mu_{test}(m))$$

(в общем случае для корректной работы сети функции и предикаты, с которыми оперирует конкретный модуль, должны быть временно заблокированы в целях недопущения нежелательных взаимоблокировок конкурирующих модулей);

$\mu_B : B \rightarrow \mathbf{P}(U_F \cup U_P) \setminus \emptyset$ – отображение, ставящее в соответствие каждому блоку подмножество совместимых элементарных обновлений интерпретации сигнатуры Σ ;

$\mu_{MB} : M \rightarrow \mathbf{P}(B) \setminus \emptyset$ – отображение, сопоставляющее каждому модулю СеАМ подмножество реализуемых им блоков;

$\mu_L : M \rightarrow \mathbf{P}(L) \setminus \emptyset$ – отображение, сопоставляющее каждому модулю подмножество проверяемых им логических условий;

C – предикат каузации, определяемый следующим образом:

$$(\forall m \in M)(\forall n \in M)(C(m, n) = p_{\text{inter}}(\mu_{\text{update}}(m), \mu_{\text{test}}(n))) ,$$

то есть некоторый модуль m_i может причинно (каузально) воздействовать на модуль n_j , в случае, когда некоторые элементы из множества предикатов и функций, модифицируемых модулем $m_i \in M$, проверяются модулем $n_j \in M$ (допускается также случай $m_i = n_j$):

$$p_{\text{inter}}(\mu_{\text{update}}(m_i), \mu_{\text{test}}(n_j)) = \begin{cases} \text{true, если } \mu_{\text{update}}(m_i) \cap \mu_{\text{test}}(n_j) \neq \emptyset, \\ \text{false, если } \mu_{\text{update}}(m_i) \cap \mu_{\text{test}}(n_j) = \emptyset. \end{cases}$$

Для сети N дополнительно задается следующая пара множеств:

Q – множество предикатов вида $q: M \rightarrow \{\text{true}, \text{false}\}$, характеризующих рабочее или нерабочее состояние модулей из множества M ; данные предикаты не могут модифицироваться никакими модулями и используются только в формулах логических условий из множества L для проверки фактов исполнения некоторых модулей в текущий момент времени;

W – множество предикатов вида $w: M \rightarrow \{\text{true}, \text{false}\}$, используемых для “внешнего” или “внутреннего” управления текущей конфигурацией сети N : в процессе функционирования сети из нее могут исключаться некоторые модули, либо они снова могут подключаться к сети; такие манипуляции с сетью могут осуществляться модулями с помощью включаемых в блоки правил обновления предикатов из множества W .

При описании операционной семантики сетей используется понятие FS-пространства, объединяющего в одно множество множества функций и предикатов. Предлагаемые абстрактные машины отличаются от других наличием причинно-следственных связей и тем, что их память состоит не из элементарных ячеек, а из некоторых структур.

При использовании алгебраических систем становится возможным сочетать как алгебраические, так и логические понятия и методы. При этом обычно рассматривается некоторый язык, алфавит которого содержит счетное число переменных $x_1, x_2, \dots$, функциональные и предикатные символы, символ равенства, логические связки, кванторы всеобщности и существования. Формулы и термы составляются по известным в многосортном исчислении предикатов первого и высших порядков правилам. В некоторых случаях в работе используется логика второго порядка, допускающая переинтерпретацию в первопорядковую логику.

Отличительной особенностью предлагаемого формализма является использование термов вида

$$(\tilde{\exists}!x \in X)p(x), (\tilde{\exists}!!x \in X)p(x), (\tilde{\forall}x \in X)p(x), (\tilde{\forall}!!x \in X)p(x) \quad (2)$$

в блоках и в α -выражениях модулей СеАМ (то есть в записи формул для логических условий в операциях α -дизъюнкции и α -итерации). Данные операторы названы квантифицированными операторами выбора кортежей из областей истинности n -арных ($n \geq 1$) предикатов. Например, в случае применения введенных операторов к бинарным предикатам соответствующие термы будем записывать в следующем виде:

$$\begin{aligned} \tilde{\exists}!(x \in X, y \in Y)q(x, y), \tilde{\exists}!!(x \in X, y \in Y)q(x, y), \tilde{\forall}(x \in X, y \in Y)q(x, y), \\ \tilde{\forall}!!(x \in X, y \in Y)q(x, y). \end{aligned} \quad (3)$$

Рассмотрим группу термов (2). Результатом выполнения оператора $\tilde{\exists}!$ является единственное значение предметной переменной x , выбранное произвольным образом из области истинности унарного предиката p . Оператор $\tilde{\exists}!!$ выбирает значение x при условии, что оно является единственным в области истинности унарного предиката p . Оператор $\tilde{\forall}$ позволяет выбрать все значения переменной x из области истинности предиката p . Оператор $\tilde{\forall}!!$ выбирает все значения переменной x из области истинности предиката p при условии, что эта область совпадает с областью определения данного предиката. Выбранные значения для обоих последних операторов составляют результирующие унарные отношения. Аналогично определяются и n -арные квантифицированные операторы выбора.

В термах вида (2), если это специально оговорено, переменная x , или переменный кортеж, аналогично реляционному исчислению кортежей может пробегать по кортежам некоторого n -арного отношения, представленного областью истинности соответствующего предиката. Для выбора i -го элемента кортежа x в этом случае необходимо использовать операцию проекции $pr_i(x)$, $i = 1, 2, \dots, n$.

Выражения с квантифицированными операторами выбора в α -условиях модулей (то есть в выражениях, заключенных в квадратные скобки) трактуются как логические условия, истинные при успешном завершении выбора и ложные – в противном случае. Вычисленные значения термов, построенных на основе квантифицированных операторов, используются в блоках, следующих за α -условиями одного и того же модуля.

В отличие от обычных термов значения термов с квантифицированными операторами выбора $\tilde{\exists}!$, $\tilde{\exists}!!$, $\tilde{\forall}$, $\tilde{\forall}!!$ не всегда могут быть вычислены, то есть, возможно, не удастся осуществить выбор ни одного кортежа из области истинности некоторого предиката. В этом случае считается, что данный оператор выполняет “тождественное” обновление R^E , то есть он не вносит ни одного изменения в интерпретацию сигнатуры Σ , а значения соответствующих предметных переменных не определены.

В выражениях для модулей сетей абстрактных машин используются элементарные правила обновления функций или предикатов вида

$$\frac{t_1, t_2, \dots, t_k, t_{k+1}}{s(t_1, t_2, \dots, t_k) \leftarrow t_{k+1}}, \quad (4)$$

где $t_1, t_2, \dots, t_k$ – термы различных сортов; s – функциональный или предикатный символ. В случае, если s – функциональный символ, то t_{k+1} суть терм любого сорта, а если s – предикатный символ, то t_{k+1} – булево выражение. Отметим, что вне рамок определения формализма машин абстрактных состояний данные правила вывода могут быть реализованы также и операциями модификации отношений в реляционных базах данных.

Определены порядок выполнения правил вида (4) при использовании в выражениях для термов квантифицированных операторов выбора, механизмы интерпретации, правила декомпозиции и модификации СеАМ, определена статическая сигнатура для представления арифметических и логических операций. Показано, что от выбора типа СеАМ и способа структуризации FS-пространства зависит сложность разрабатываемого аппаратно-программного обеспечения. Даны примеры построения исполняемых формальных спецификаций на основе выражений для модулей сети СеАМ. Подкласс сетей СеАМ, при определении которого используются межмодульные темпоральные связи, а также обе управляющие операции – α -дизъюнкция и α -итерация – используется также под названием РСеАМ (расширенные сети абстрактных машин).

Введено понятие “активной” семантической сети (АСеС) как варианта сценарной сети (или просто сценария), построение которой, в отличие от известных в области искусственного интеллекта сценарных моделей, основано на алгебре активных семантических сетей, в которой исходные (“элементарные”) концептуальные графы и логические условия составляют систему образующих. Решена задача определения самомодифицируемых сценариев функционирования систем и сетей хранения и обработки данных сетями абстрактных машин.

Модифицируя бинарные темпоральные предикаты, можно в существенной степени манипулировать свойствами сценариев. Например, можно изменять степень параллелизма при выполнении тех или иных действий, предусмотренных сценарием. Рассмотрены временные сценарии и определены правила формирования запросов к сценариям. На примере сети хранения и обработки данных определены правила эволюции семантических сетей и концептуальных графов. Даны рекомендации по интерпретации сценарных сетей сетями абстрактных машин, включающие правила определения темпоральных операций, правила синхронизации сценариев. Определены основные алгебраические свойства темпоральных операций. Приведены примеры иерархического и одноуровневого представления сценариев, а также их формульные и графические интерпретации.

На основе сценарных представлений и интерпретирующих СеАМ формализованы процессы централизованного и децентрализованного управления размещением ресурсов в сетях хранения и обработки данных. Приведены примеры логико-алгебраических спецификаций пространственных операций. Проанализированы особенности определения и применения содержательных и формальных методов в проектировании систем и сетей хранения и обработки данных. Даны предварительные предложения по реализации формальных моделей на основе современных инструментальных средств и технологий.

Достоинством предлагаемых методов формализации является возможность их использования при теоретическом обосновании методов построения концептуальных, математических и алгоритмических моделей систем и сетей хранения и обработки данных, обеспечивающих сокращение затрат на создание моделей, их настройку и обеспечение адекватности проектируемой системы. Предлагаемый подход к формализации, моделированию и проектированию систем и сетей хранения и обработки данных базируется на унифицированном описании взаимодействия аппаратного и программного обеспечения. При этом структурные связи между аппаратными и программными модулями в логико-алгебраических моделях задаются предикатами и функциями, а логические связи – замкнутыми выражениями в исчислении предикатов.

Процесс конструирования имитационных программ для работы с моделями систем аналогичен процессу конструирования системных программ, управляющих реальными процессами и ресурсами в данных системах при реализации функций информационного обмена, хранения и обработки информации.

В качестве основной математической модели иерархического проектирования предложено использовать иерархические алгебраические системы. Иерархическое проектирование на каждом уровне сопровождается сильным обогащением многоосновной алгебраической системы соответствующего уровня. Определена иерархия (направленное множество) эволюционирующих многоосновных алгебраических систем

$$\mathbf{A} = \langle A_1, A_2, \dots, A_n \rangle$$

с соответствующими вектором сигнатур

$$\Sigma = \langle \Sigma_1, \Sigma_2, \dots, \Sigma_n \rangle,$$

вектором множеств основ

$$\mathbf{A} = \langle A_1, A_2, \dots, A_n \rangle$$

и вектором интерпретаций сигнатур

$$\mathbf{I}^{\mathbf{A}} = \langle I^{A_1}, I^{A_2}, \dots, I^{A_n} \rangle.$$

От известного направленного множества алгебраических систем данное множество отличается тем, что здесь все системы могут иметь различную сигнатуру, причем мощность каждой последующей системы больше или

равна предыдущей. Иерархическая система **A** функционирует, переходя от одного вектора $\mathbf{I}^A(t_i)$ интерпретаций сигнатур к другому вектору $\mathbf{I}^A(t_{i+1})$, $t_{i+1} > t_i$, $t_0 \geq 0$, при этом текущий вектор интерпретаций сигнатур $\mathbf{I}^A(t)$ соответствует текущему абстрактному состоянию иерархии алгебраических систем; на каждом уровне элементы вектора интерпретации сигнатур изменяются согласованно друг с другом. Все уровни модели при этом остаются различимыми (видимыми) для пользователя, что позволяет в дальнейшем построить иерархию сети абстрактных машин и затем перейти к ее реализации иерархической сетью виртуальных машин. В зависимости от интерпретации иерархическая система **A** может использоваться как основа для формализации процессов иерархического проектирования, так и для проектирования иерархических систем.

Определенные во второй главе языки сетей абстрактных машин, временных и пространственных сценариев обладают в совокупности свойствами, позволяющими поддерживать концепцию объектно-ориентированной архитектуры, являющейся дальнейшим развитием виртуальной архитектуры путем ее дальнейшей функциональной ориентации. Данные языки используют единое представление для всех объектов и содержат формализованные команды высокого уровня для управления объектами. Реализация данных языков программно-аппаратными средствами основана на интеграции средств управления процессами, виртуальной памятью и данными.

В третьей главе предлагается методология концептуального и имитационного поведенческого моделирования систем и сетей внешнего хранения и обработки данных, обеспечивающая иерархическое проектирование формально определенных моделей, отличающихся адекватным представлением процессов управления информационным обменом, хранением и обработкой информации.

Разрабатываются методы и средства поддержки функционально-структурной и функционально-топологической декомпозиции архитектурных моделей систем и сетей хранения и обработки данных. Данные модели и методы соответствуют современной технологической инфраструктуре параллельных и распределенных баз данных, кластерных и мультипроцессорных систем, сетевых соединений, интеллектуальных контроллеров и накопителей информации.

Предложена методология формализованного иерархического синтеза сценарных моделей, базирующаяся на алгебре концептуальных графов. Здесь одним множеством концептов является множество событий, а другое множество концептов представляет объекты предметной области – сообщения, блоки данных, файлы, узлы хранения или обработки данных, линии связи, причем события связаны с объектами ролевыми отношениями, а между собой – темпоральными и управляющими отношениями, что позволяет определить основы схематологии концептуальных сценарных моделей

и расширить границы их применимости в процессе многоуровневого проектирования систем и сетей внешнего хранения и обработки данных.

Предложено указанные подходы распространить и на представление операционной семантики проектируемой системы. Общие черты проектирования подобных систем предлагается использовать при формировании унифицированного архитектурного стиля и языков описания архитектуры систем и сетей хранения и обработки данных.

Для конструирования сценариев и выполнения над ними разнообразных процедур модификации используются сети абстрактных машин. Данное предложение обусловлено начальной неинтерпретированностью формализма сетей абстрактных машин. Для формализации данных понятий над иерархическим сценарием A определена суперсеть абстрактных машин SN . Суперсеть включает в свой состав модули, выполняющие некоторые модификации сценария A . Определяется также сценарий, представленный в реляционной форме, или просто реляционный сценарий. Реляционный сценарий C_R определен как набор:

$$C_R = (S, R, Op, U_R, U_{Op}, SN),$$

где S – множество подсценариев, возможно, составных; в этом множестве выделяются подмножества S_0 и S_F – начальных и конечных подсценариев;

R – множество предикатов структуризации (формирования) сценария, в которое могут быть включены бинарные темпоральные предикаты, тернарный предикат Op_v и бинарный предикат $Op_{\{ \}$;

Op – множество операций, одноименных предикатам из множества R ;

U_R – множество правил обновления предикатов из множества R ;

U_{Op} – множество правил обновления операций из множества Op ;

SN – управляющая суперсеть.

Предложен метод интерпретации концептуальных графов и самомодифицируемых сценариев сетями абстрактных машин, основанный на реализации базовых сценарных операций модулями-продукциями и модулями-процедурами сетей абстрактных машин.

На основе описанных представлений предложена технология согласования процессов и объектов, или технология активной классной доски, предназначенная для моделирования и реализации в сетевой среде систем и сетей хранения и обработки данных. Данная технология основана на агентно-ориентированных моделях и реализуемого на базе сетей абстрактных машин объектно-ориентированного проектирования. Основным отличием предлагаемой технологии является унифицированное представление декларативных и процедурных знаний, хранимых в разделяемом пространстве памяти, что сокращает затраты на создание модели и проекта системы, в том числе за счет их адекватности. Идентичность модели и проекта достигается за счет того, что по мере развития формализуемых концептуальных моделей в результате, после проведения экспериментов с результирующей

имитационной поведенческой моделью, ее логико-алгебраические спецификации становится возможным использовать в качестве спецификаций для рабочего проекта. С другой стороны, положительной особенностью данной технологии является возможность использования других известных технологий, связанных с проектированием систем искусственного интеллекта, а также технологии объектно-ориентированного проектирования систем.

Предложены и реализованы практически в сетевой среде архитектурные концептуальные схемы реализации технологии активной "классной доски": "мобильные агенты – стационарные функции" и "стационарные агенты – перемещаемые функции". Данные схемы реализуют архитектуру сетей хранения и обработки данных с виртуальной памятью, ядром которых является система хранения данных. Даны примеры представления и синтеза сценариев, интерпретируемых сетями абстрактных машин, а также механизмы их реализации.

Допустим, что необходимо сформировать пул ресурсов хранения с группой серверов, образующих множество C , с назначаемыми им дисковыми накопителями из множества D с соответствующими интеллектуальными контроллерами из множества U . В подобной системе можно легко переназначать ресурсы подсистемы хранения данных между серверами. Сеть содержит два коммутатора – K_1 и K_2 , на основе первого из которых сформирована абонентская сеть, связывающая серверы с клиентскими станциями, а на основе второго – собственно сеть хранения данных с дисковыми модулями и интеллектуальными контроллерами. В сети ВЗУ на базе коммутатора K_2 обеспечиваются переключения дисковых модулей между серверами. Пусть Z – множество клиентских станций (или просто "клиентов"). Пусть также $z_i \in Z$, $i = 1, 2$; $c_j \in C$, $j = 1, 2, 3, 4$; $d_k \in D$, $k = 1, 2, \dots, 16$. Работа сети хранения данных при выполнении операции записи для двух клиентских станций может быть описана следующими системой из $2 \times 3 = 6$ выражений для модулей СеАМ (по три модуля на каждую клиентскую станцию):

$$\begin{aligned} m_1^{(i)} &= [p_{Client}(z_i)]([\exists!c)p_{Server}(c)][[\exists!d)p_{Disk}(d)](\{p_{Client}(z_i) \leftarrow false, \\ p_{Com}(z_i, c, d) &\leftarrow true, p_{Server}(c) \leftarrow false, p_{Disk}(d) \leftarrow false, \\ p_{Delay}(z_i) &\leftarrow true\} \vee R^E) \vee R^E) \vee R^E); \\ m_2^{(i)} &= [p_{Delay}(z_i)](\{m_{Work}(z_i), p_{Delay}(z_i) \leftarrow false, p_{End}(z_i) \leftarrow true\} \vee R^E); \\ m_3^{(i)} &= [p_{End}(z_i) \& \exists!(z_i, c, d)p_{Com}(z, c, d)](\{p_{Com}(z_i, c, d) \leftarrow false, \\ p_{Server}(c) &\leftarrow true, p_{Disk}(d) \leftarrow true, p_{End}(z_i) \leftarrow false, p_{Client}(z_i) \leftarrow true\} \vee R^E), \end{aligned}$$

где p_{Client} , p_{Server} , p_{Disk} – унарные предикатные символы, характеризующие активности клиента, сервера и дискового модуля (с контроллером) соответственно; p_{Delay} – унарный предикатный символ, используемый для задания связи модуля $m_1^{(i)}$ с модулем задержки $m_2^{(i)}$; p_{Com} – тернарный предикатный символ, используемый для задания связи клиента, сервера и дискового мо-

дуля в процессе выполнения рабочей операции Work; $m_{\text{Work}}(z_i)$ – модуль, задающий действия, выполняемые при реализации рабочей операции Work для заданного клиентского запроса z_i ; p_{End} – унарный предикатный символ, используемый при описании связи модуля задержки $m_2^{(i)}$ с завершающим операцией модулем $m_3^{(i)}$; c и d – предметные переменные, пробегающие по элементам соответствующих множеств C и D . Множество устройств U здесь и далее не учитывается, так как каждый дисковый модуль имеет собственный интеллектуальный контроллер, время занятости которого совпадает с временем занятости модуля. В общем случае здесь для описания функционирования системы хранения данных достаточно $q \times 3$ выражений для модулей СеАМ, где q – число клиентских станций (или типов запросов).

Приведенные выражения для модулей представляют собой замкнутые выражения (без свободных переменных) в алгебре модулей СеАМ. Переходя к выражениям для модулей СеАМ, не содержащим вхождений квантифицированных операторов выбора $\exists!$ и тернарного предикатного символа p_{Com} , получим систему из $2 \times 4 \times 16 \times 3 = 384$ выражений (при $i = 1, 2$; $j = 1, 2, 3, 4$; $k = 1, 2, \dots, 16$):

$$\begin{aligned} m_1^{(i)} &= [p_{\text{Client}}(z_i) \& p_{\text{Server}}(c_j) \& p_{\text{Disk}}(d_k)](\{ p_{\text{Client}}(z_i) \leftarrow \text{false}, \\ p_{\text{Server}}(c_j) &\leftarrow \text{false}, p_{\text{Disk}}(d_k) \leftarrow \text{false}, p_{\text{Delay}}(z_i) \leftarrow \text{true} \} \vee R^E); \\ m_2^{(i)} &= [p_{\text{Delay}}(z_i)](\{ m_{\text{Work}}(z_i), p_{\text{Delay}}(z_i) \leftarrow \text{false}, p_{\text{End}}(z_i) \leftarrow \text{true} \} \vee R^E); \\ m_3^{(i)} &= [p_{\text{End}}(z_i)](\{ p_{\text{Server}}(c_j) \leftarrow \text{true}, p_{\text{Disk}}(d_k) \leftarrow \text{true}, p_{\text{End}}(z_i) \leftarrow \text{false}, \\ p_{\text{Client}}(z_i) &\leftarrow \text{true} \} \vee R^E). \end{aligned}$$

Без предиката p_{Com} здесь для задания каждой связи “клиент – сервер – дисковый модуль” используется пара модулей – один ($m_1^{(i)}$) в начальной фазе операции, а другой ($m_3^{(i)}$) – в конечной. В данных выражениях справа от предикатных символов в скобках приведены обозначения (константы) для конкретных объектов, то есть здесь используются не предикаты, а только высказывания.

При $i = 1, 2, \dots, q$; $j = 1, 2, \dots, m$; $k = 1, 2, \dots, n$, в общем случае для спецификации системы хранения данных без использования квантифицированного оператора $\exists!$ и тернарного предикатного символа p_{Com} потребуется $q \times m \times n \times 3$ выражений для соответствующих модулей СеАМ. Например, при $q=2$, $m=8$, $n=16$ потребуется 768 модулей, а при $q=2$, $m=8$, $n=64$ – 3072 модуля. При использовании же связывающего предиката p_{Com} и оператора $\exists!$ потребуется всего шесть модулей, однако в этом случае потребуется также дополнительная память для хранения области истинности предиката p_{Com} .

Логико-алгебраические спецификации данной системы, рассматриваемой как одна из базовых, позволяют упростить создание нового аппаратного, микропрограммного и программного обеспечения систем хранения и

обработки данных на принципах согласования и координации процессов и объектов.

Четвертая глава посвящена функциональной организации интеллектуальных систем управления внешней памятью ЭВМ и систем, основанной на моделях и методах согласования и координации процессов и объектов. Выбранные модели и методы базируются на сценарных представлениях и сетях абстрактных машин, используемых при проектировании развитых систем управления внешней памятью (СУВП) ЭВМ. Рассматриваются также вопросы проектирования интеллектуальных систем внешней памяти (ИСВП) ЭВМ на основе функционального подхода, учитывающего состав и взаимосвязь функций системы. Используется двойственная трактовка термина “интеллектуализация”: как развитие функциональных возможностей системы и как использование методов искусственного интеллекта при ее реализации. Обосновывается построение концептуальных и имитационных поведенческих моделей в процессе проектирования систем и сетей внешнего хранения и обработки данных и синтеза функционально и топологически централизованных и децентрализованных структур.

Выделены следующие укрупненные этапы синтеза сложных систем управления хранением и обработкой данных:

- на первом этапе осуществляется концептуализация целей управления и выбор функциональных блоков системы, определяется функционально-целевая системная информация о составе компонентов и систем и причинно-следственная информация об их взаимодействиях. На этой основе определяются состав сетевых (в том числе на базе парадигм систем искусственного интеллекта и ситуационного управления) и логико-алгебраических спецификаций, определяются уровни функционально-топологической интеграции или декомпозиции;

- на втором этапе синтеза определяются топология, структура, концептуально-формализованное описание отношений между подсистемами, составляющими понятия предметной области. На этом этапе структурные связи между понятиями предметной области задаются сигнатурой, сорта которой интерпретируются как множества, а логические связи между понятиями задаются предикатными выражениями, записываемыми в этой сигнатуре;

- на третьем этапе определяются и специфицируются процедурные знания о функционировании систем управления хранением и обработкой данных, производится анализ построенной поведенческой модели системы управления;

- на четвертом этапе производится анализ окончательного варианта реализации системы и осуществляется реализация самой системы управления хранением и обработкой данных.

Проведена функционально-структурная декомпозиция систем управления внешним хранением и обработкой данных, позволившая упростить архитектурное проектирование и сократить сроки разработки. В основу декомпозиции положен анализ пространственно-временных свойств процессов, управляющих передачей и обработкой данных.

Для реализации функциональной структуры на микропроцессорной основе в виде мультимикропроцессорной системы с непосредственными связями необходимо произвести разбиение множества функциональных модулей (ФМ) на так называемые реализационные группы. Функциональные модули одной реализационной группы реализуются на отдельной единице реализации, в качестве которой может выступать микропроцессорная система, а также ее отдельные компоненты: процессор P , память M и интерфейсное оборудование I . Графическое представление данной структуры назовем МРІ-диаграммой.

Все ФМ одной реализационной группы объединяются в единый ФМ, для которого и определяется единица реализации – конструктивный модуль. Пусть для некоторой СУВП представлена структура аппаратных средств, полученная в результате трансформации функциональной структуры, причем в данном случае выделены следующие реализационные группы: FM_1 ; FM_3 ; $(FM_{2.2}, FM_{2.3})$; $FM_{2.4}$; $(FM_{2.5}, FM_{2.6}, FM_{2.7})$. Считается, что ФМ низшего уровня иерархии, подчиненные ФМ верхнего уровня, входят в ту же самую реализационную группу.

Отношение развертывания здесь задаётся областью истинности бинарного предиката D вида:

$$D: K \times F \rightarrow \{\text{true}, \text{false}\},$$

где K – множество активных элементов (процессорных и интерфейсных элементов конструктивных модулей); F – множество функциональных модулей. Область истинности предиката D формируется модулем CeAM M_{start} , описываемым следующим логико-алгебраическим выражением:

$$\begin{aligned} M_{\text{start}} = [\alpha_{\text{start}}](&\{D(I_2, FM_1) \leftarrow \text{true}, D(P_1, FM_{2.1}) \leftarrow \text{true}, D(P_2, FM_{2.2}) \leftarrow \text{true}, \\ &D(P_2, FM_{2.3}) \leftarrow \text{true}, D(P_3, FM_{2.4}) \leftarrow \text{true}, D(P_4, FM_{2.5}) \leftarrow \text{true}, \\ &D(P_4, FM_{2.6}) \leftarrow \text{true}, D(P_4, FM_{2.7}) \leftarrow \text{true}, D(I_3, FM_3) \leftarrow \text{true}\} \vee R^E). \end{aligned}$$

По завершении выполнения модуля CeAM, реализующего данное выражение, функциональные модули размещаются на конструктивных модулях СУВП. Здесь α_{start} – признак готовности системы к загрузке. Выполнение данного модуля CeAM в реальной системе сопровождается загрузкой соответствующего программного обеспечения в основную память процессорных модулей, составляющих СУВП.

Предложено при построении концептуальных моделей функционально-структурной организации многофункциональных многопроцессорных и сетевых систем управления внешним хранением и обработкой данных использовать естественный параллелизм процессов и механизмы логического

вывода, что позволяет повысить эффективность последующей аппаратно-программной реализации систем с развитым интеллектом.

Разработаны математические модели функционально-централизованных и функционально-децентрализованных систем и сетей внешнего хранения и обработки данных, основанные на формально определенных сценариях и сетях абстрактных машин. В отличие от известных моделей, данные формальные модели использованы не только для построения формально определенных имитационных поведенческих моделей, но и как непосредственно исполняемые формальные спецификации, что повышает качество и сокращает сроки разработки системы, в том числе ее системного и сетевого программного обеспечения.

Рассмотрим централизованную и децентрализованную схемы управления выполнением сценария F_0 для некоторой СУВП:

$$F_0 = F_1 \uparrow ((F_2 \downarrow F_3) | (F_4 \uparrow [\alpha] (F_5 \vee A^E) \uparrow F_6)) \uparrow (F_7 \parallel F_8) \uparrow F_k,$$

где “ $\uparrow$ ”, “ $\downarrow$ ”, “ $|$ ” и “ $\parallel$ ” – символы темпоральных операций “непосредственного следования”, “неодновременного выполнения”, “возможно одновременного выполнения” и “перекрытия интервалов выполнения” подсценариев соответственно, A^E – тождественный подсценарий, а α – некоторое условие.

Централизованная схема реализуется под управлением выделенного процессора P_0 . Выполнение каждого функционального модуля, или подсценария, включает три фазы: начальную, рабочую и конечную. Начальная и конечная фазы некоторого подсценария F_i ($i = 1, 2, \dots, 8$) при централизованном управлении реализуются начальным и конечным модулями СеАМ m_{bi} и m_{ei} соответственно под управлением процессора P_0 . Начальный модуль m_{bi} инициирует выполнение подсценария F_0 , передавая управляющее воздействие в соответствующий физический процессор $P(F_i)$ путем выполнения правила обновления предиката $d_{bi}(d_i) \leftarrow \text{true}$, где d_{bi} – унарный предикат, а d_i – имя агента-“демона”, выполняющего рабочую фазу подсценария F_i . Запись $P(F_i)$ обозначает процессор, на котором реализуется ФМ F_i . Реализуя конечную фазу выполнения сценария F_i , “демон” d_i завершает работу, выполняя правило обновления предиката $d_{ei}(d_i) \leftarrow \text{true}$. Истинность высказывания $d_{ei}(d_i)$ проверяется в условной части модуля m_{ei} , который отмечает завершающую фазу сценария F_i выполнением правила обновления предиката $d_{ei}(d_i) \leftarrow \text{false}$.

Агент a_0 , интерпретирующий на процессоре P_0 управляющую СеАМ по централизованной схеме, выполняет переходы от одного модуля СеАМ к другому, реализуя сценарий в целом. Каждый модуль m_{bi} начинает свою работу, проверяя стартовое условие, а каждый модуль m_{ei} , завершая свою работу, передает управление одному или нескольким модулям, обновляя один или несколько каузальных предикатов, с помощью которых реализуется причинное воздействие данного модуля на другие. Система логико-алгебра-

ических выражений для модулей, реализующих сценарий F_0 по схеме с централизованным управлением, имеет следующий вид:

для подсценария F_1 :

$$m_{b1} = [p_{b1}(a_0)](\{p_{b1}(a_0) \leftarrow \text{false}, p_1(a_0) \leftarrow \text{true}, d_{b1}(d_1) \leftarrow \text{true}\} \vee R^E);$$

$$m_{e1} = [p_1(a_0) \& d_{e1}(d_1)](\{p_1(a_0) \leftarrow \text{false}, d_{e1}(d_1) \leftarrow \text{false},$$

$$p_{b2}(a_0) \leftarrow \text{true}, p_{b3}(a_0) \leftarrow \text{true}, p_{b4}(a_0) \leftarrow \text{true}\} \vee R^E);$$

для подсценария F_2 :

$$m_{b2} = [p_{b2}(a_0) \& p_{2,3}(r_0)](\{p_{b2}(a_0) \leftarrow \text{false}, p_{2,3}(r_0) \leftarrow \text{false},$$

$$p_2(a_0) \leftarrow \text{true}, d_{b2}(d_2) \leftarrow \text{true}\} \vee R^E);$$

$$m_{e2} = [p_2(a_0) \& d_{e2}(d_2)](\{p_2(a_0) \leftarrow \text{false}, d_{e2}(d_2) \leftarrow \text{false}, p_{2,3}(a_0) \leftarrow \text{true},$$

$$p_{e2}(a_0) \leftarrow \text{true}\} \vee R^E);$$

для подсценария F_3 :

$$m_{b3} = [p_{b3}(a_0) \& p_{2,3}(r_0)](\{p_{b3}(a_0) \leftarrow \text{false}, p_{2,3}(r_0) \leftarrow \text{false}, p_3(a_0) \leftarrow \text{true},$$

$$d_{b3}(d_3) \leftarrow \text{true}\} \vee R^E);$$

$$m_{e3} = [p_3(a_0) \& d_{e3}(d_3)](\{p_3(a_0) \leftarrow \text{false}, d_{e3}(a_0) \leftarrow \text{false}, p_{2,3}(r_0) \leftarrow \text{true},$$

$$p_{e3}(a_0) \leftarrow \text{true}\} \vee R^E);$$

для подсценария F_4 :

$$m_{b4} = [p_{b4}(a_0)](\{p_{b4}(a_0) \leftarrow \text{false}, p_4(a_0) \leftarrow \text{true}, d_{b4}(d_4) \leftarrow \text{true}\} \vee R^E);$$

$$m_{e4} = [p_4(a_0) \& d_{e4}(d_4)](\{p_4(a_0) \leftarrow \text{false}, d_{e4}(d_4) \leftarrow \text{false}, p_{e4}(a_0) \leftarrow \text{true}\} \vee R^E);$$

для подсценария F_5 :

$$m_{b5} = [\alpha \& p_{e4}(a_0)](\{p_{e4}(a_0) \leftarrow \text{false}, p_5(a_0) \leftarrow \text{true}, d_{b5}(d_5) \leftarrow \text{true}\} \vee R^E);$$

$$m_{e5} = [p_5(a_0) \& d_{e5}(d_5)](\{p_5(a_0) \leftarrow \text{false}, d_{e5}(d_5) \leftarrow \text{false}, p_{b6}(a_0) \leftarrow \text{true}\} \vee R^E);$$

вспомогательный модуль:

$$m_{4,6} = [\neg \alpha \& p_{e4}(a_0)](\{p_{e4}(a_0) \leftarrow \text{false}, p_{b6}(a_0) \leftarrow \text{true}\} \vee R^E);$$

для подсценария F_6 :

$$m_{b6} = [p_{b6}(a_0)](\{p_{b6}(a_0) \leftarrow \text{false}, p_6(a_0) \leftarrow \text{true}, d_{b6}(d_6) \leftarrow \text{true}\} \vee R^E);$$

$$m_{e6} = [p_6(a_0) \& d_{e6}(d_6)](\{p_6(a_0) \leftarrow \text{false}, d_{e6}(d_6) \leftarrow \text{false}, p_{6,7}(a_0) \leftarrow \text{true}\} \vee R^E);$$

для подсценария F_7 :

$$m_{b7} = [p_{e2}(a_0) \& p_{e3}(a_0) \& p_{6,7}(a_0)](\{p_{e2}(a_0) \leftarrow \text{false}, p_{e3}(a_0) \leftarrow \text{false},$$

$$p_{6,7}(a_0) \leftarrow \text{false}, p_7(a_0) \leftarrow \text{true}, d_{b7}(d_7) \leftarrow \text{true}, p_{7,8}(a_0) \leftarrow \text{true}\} \vee R^E);$$

$$m_{e7} = [p_7(a_0) \& d_{e7}(d_7) \& p_{8,7}(a_0)](\{p_7(a_0) \leftarrow \text{false}, d_{e7}(d_7) \leftarrow \text{false}, p_{8,7}(a_0) \leftarrow \text{false},$$

$$p_k(a_0) \leftarrow \text{true}\} \vee R^E);$$

для подсценария F_8 :

$$m_{b8} = [p_{7,8}(a_0)](\{p_{7,8}(a_0) \leftarrow \text{false}, p_8(a_0) \leftarrow \text{true}, d_{b8}(d_8) \leftarrow \text{true}\} \vee R^E);$$

$$m_{e8} = [p_8(a_0) \& d_{e8}(d_8)](\{p_8(a_0) \leftarrow \text{false}, d_{e8}(a_0) \leftarrow \text{false},$$

$$p_{8,7}(a_0) \leftarrow \text{true}\} \vee R^E).$$

В приведенных выше выражениях предикаты p_i ($i = 1, 2, \dots, 8$) связывают начальные и конечные модули CeAM, соответствующие каждому подсцена-

рию F_1 , а символ a_0 именуется агент-сервер, реализующий на процессоре P_0 поиск и запуск готовых к выполнению описанных выше модулей управляющей сети СеАМ. Остальные предикаты используются для управления запуском модулей. Символ r_0 обозначает неразделяемый ресурс, используемый подсценариями F_2 или F_3 , поэтому данные подсценарии не могут выполняться одновременно. Программы, реализующие все модули (“методы” агента a_0), здесь размещены вместе с кодом самого агента в локальной памяти процессора P_0 . Предполагается, что процессор P_0 имеет доступ к модулям основной памяти всех остальных процессоров для организации и совместного использования FS-пространства, в котором хранятся информационные объекты, представляющие предикаты – объектные (используемые для запуска агентов-демонов) и каузальные (используемые для связывания модулей управляющей сети).

Реализация децентрализованной схемы управления в СУВП не требует наличия специально выделенного процессора; здесь программы, соответствующие модулям управляющей сети (“методы” соответствующих агентов-серверов $a_1, a_2, \dots, a_8$), размещаются в локальной или основной памяти процессоров, реализующих подсценарии $F_1, F_2, \dots, F_8$ основного сценария F_0 . Коды агентов-демонов $d_1, d_2, \dots, d_8$ для обеих схем управления размещаются в локальной или основной памяти функциональных процессоров. Логико-алгебраические выражения для модулей децентрализованной сети СеАМ отличаются от приведенных выше выражений использованием имен агентов-серверов в качестве значений аргументов для унарных предикатов.

На основании методов, предложенных во второй и третьей главах, определены системная и сетевая, виртуальная и физическая инфраструктуры как платформы для реализации абстрактных вычислительных моделей. Для описания данных платформ использованы не только формальные методы, но и неформальные, основанные на диаграммных макроопределениях и МРІ-диаграммах, упрощающие начальные этапы функционально-структурного проектирования.

Расширены возможности сетевой виртуализации и интеграции функций и сервисов хранения данных, позволяющих организовать обработку структурированной информации средствами виртуального сетевого процессора базы данных в произвольной системе или сети внешнего хранения и обработки данных с виртуальной памятью достаточного объема.

Разработаны технологические комплексы программ, составляющие подсистему синтеза структур и программного обеспечения, а также подсистему спецификации и верификации систем управления внешним хранением и обработкой данных. Данные комплексы основаны на использовании формальных и неформальных моделей; использование логико-алгебраических методов в описании систем, алгебры сценариев, алгебры абстрактных машин и алгебры развертывания структур позволяет строить иерархически организованное прикладное и системное программное обеспечение систем и сетей хранения и обработки данных.

В пятой главе рассмотрены вопросы функционально-топологической децентрализации в архитектурном проектировании систем и сетей внешнего хранения и обработки данных на основе использования формальных представлений распределенных процессов и объектов, взаимодействующих через общее структурированное пространство памяти.

Исследовано применение новых формализмов для описания согласованных, или скоординированных взаимодействий процессов и объектов, базирующихся на декларативном и процедурном подходах к представлению знаний о функционировании распределенных систем. Использован также алгебраический подход к определению операционной семантики распределенных систем хранения и обработки данных, таких как сети многофункциональных внешних запоминающих устройств и сетевые процессоры баз данных, основанный на определении данных систем сетями абстрактных машин.

Обоснована необходимость разработки новых методов и средств поддержки функционально-структурной и функционально-топологической декомпозиции архитектурных моделей систем и сетей хранения и обработки данных. Данные модели и методы должны соответствовать современной технологической инфраструктуре параллельных и распределенных баз данных, кластерных и мультипроцессорных систем, сетевых соединений, интеллектуальных контроллеров и накопителей информации.

Предложены принципы построения систем управления применительно к объектам, функционирование которых представимо в классе сетей абстрактных машин. На основе логико-алгебраического подхода, формальных и неформальных моделей и методов искусственного интеллекта, теории иерархически структурированных абстрактных машин и сценарных представлений предложена методология построения виртуализированных систем и сетей внешнего хранения и обработки данных, узлы которых сочетают функции виртуализации с функциями обработки структурированных данных.

Расширены возможности сетевой виртуализации на основе модели сетей абстрактных машин, что позволяет интегрировать функции и сервисы внешнего хранения и обработки данных за счет эффективного использования структурированной виртуальной памяти как для хранения обрабатываемых данных, так и для хранения “каузальной” информации, посредством которой объекты и процессы согласуют свои действия.

Разработаны новые модели управления распределенными ресурсами, включающие концептуальные модели и формально определенные имитационные поведенческие модели, отличающиеся от известных тем, что на начальном и промежуточном этапах они описываются семантическими сетями (в том числе семантическими сетями с событиями – сценариями), формализуемыми сетями абстрактных машин, а на завершающих этапах интерпретируются сетями алгоритмических модулей.

Разработаны новые формальные модели организации функционирования распределенного ресурса – многофункциональной системы внешней памяти, подключаемой к вычислительной сети. В отличие от известных, в данных моделях эффективно реализуются взаимодействия компонент, что позволяет разработчику модифицировать сложные модели, получая новые качественные результаты. Рассмотрены подобные трансформации формальных моделей, позволившие перейти от централизованной архитектуры систем хранения и обработки данных к децентрализованной (сетевой).

В сетях ВЗУ совокупность сетевых дисковых модулей может быть представлена некоторым коллективно используемым ресурсом R . Доступ к подобному ресурсу с несколькими единицами может быть осуществлен с помощью запросов следующих видов: e -запрос требует для исполнения все единицы ресурса R , s_1 -запрос требует одну определенную единицу данного ресурса, а s_2 -запрос требует одну произвольную единицу ресурса. Подобного рода запросы характерны, например, для распределенных RAID-массивов (RAID – Redundant Array of Independent Disks – избыточный массив независимых дисков) и реплицированных распределенных баз данных, реализуемых на базе сетей ВЗУ.

Выполнение e -запроса заключается в том, что некоторый процесс-агент (далее просто агент), реализуя событие p , запрашивает у агентов-менеджеров специальные “жетоны” на право использования всех единиц ресурса R – узлов сети ВЗУ, по одному жетону t на каждую единицу u . После получения всех жетонов соответствующие единицы ресурса блокируются для данного запроса и он выполняется. После использования агент возвращает жетоны, и освобожденные единицы ресурса могут использоваться другими агентами. Пусть T – множество жетонов $t \in T$, Y – множество узлов сети $u \in Y$, причем $|T| = |Y|$, P – множество процедур (фактов) обработки запросов $p \in P$, e_1 – текущий e -запрос. На концептуальном графе отношение принадлежности индивида абстрактному понятию “тип” представлено именем предиката *Конкр.* (от слова “конкретизация”), используются также предикатные имена и соответствующие им связывающие узлы *Это* (для представления отношения между первым типом и более общим вторым типом), *Элем.* (от слова “элемент”, для представления отношения принадлежности элемента множеству), *Подмн.* (от слова “подмножество”, для представления отношения между двумя множествами, где первое множество является подмножеством другого), *Имеет* (для представления отношения принадлежности). В выражениях для модулей СеАМ мы используем соответствующие перечисленным именам бинарные предикатные символы $f_{\text{Конкр.}}$, $f_{\text{Это}}$, $f_{\text{Элем.}}$, $f_{\text{Подмн.}}$ и $f_{\text{Имеет}}$. Используются также бинарные предикатные символы $f_{\text{Агент}}$, $f_{\text{Разр.}}$, $f_{\text{Объект}}$, а также унарные предикатные и функциональные символы, описанные ниже.

В связи с тем, что логические формы описания концептуальных графов носят непроцедурный характер, концептуальный граф описывается не логической формой, а СеАМ-выражением следующего вида:

$$m_e = [(\forall!!t \in T)f_T(t) \& f_E(e_1)] [(\forall!!y \in Y)f_Y(y)] [(\exists!p \in P)f_P(p)] \\ (\{f_{\text{Агент}}(p, e_1) \leftarrow \text{true}, f_{\text{Разр.}}(p, t) \leftarrow \text{true}, f_{\text{Объект}}(p, y) \leftarrow \text{true}, \\ f_{\text{Конкр.}}(p, \text{Обработка}) \leftarrow \text{true}, f_{\text{Конкр.}}(y, \text{Узел}) \leftarrow \text{true}, \\ f_{\text{Конкр.}}(t, \text{Жетон}) \leftarrow \text{true}, f_{\text{Конкр.}}(e_1, \text{Запрос}) \leftarrow \text{true}, \\ f_P(p) \leftarrow \text{false}, f_Y(y) \leftarrow \text{false}, f_T(t) \leftarrow \text{false}, f_E(e_1) \leftarrow \text{false}\} \vee R^E) \vee R^E) \vee R^E).$$

В процессе выполнения модуля m_e происходит выбор нужных объектов и конкретизация отношений (предикаты конкретизации типов объектов на графе не показаны). Порядок вычисления и проверки условий в модуле m_e задается процедурой выполнения запроса: при наличии всех жетонов и готовности всех узлов сети ВЗУ начинается обслуживание е-запроса.

Здесь и далее S – множество запросов; $E \subseteq S$ – подмножество запросов, каждый из которых требует все единицы ресурса R ; $S_1 \subseteq S$ – подмножество запросов, требующих одну предопределенную единицу ресурса R ; $S_2 \subseteq S$ – подмножество запросов, требующих одну произвольную единицу ресурса R ; $I = \{1, 2, \dots, n\}$ – конечное множество индексов, такое, что $|I| = |T| = |Y|$.

В выражении для модуля m_e используются следующие бинарные предикаты:

$$f_{\text{Конкр.}}: (P \cup T \cup Y) \times \{\text{Обработка}, \text{Жетон}, \text{Узел}\} \rightarrow \{\text{true}, \text{false}\}; \\ f_{\text{Агент}}: P \times (E \cup S_1 \cup S_2) \rightarrow \{\text{true}, \text{false}\}; \\ f_{\text{Разр.}}: P \times T \rightarrow \{\text{true}, \text{false}\}; \\ f_{\text{Объект}}: P \times Y \rightarrow \{\text{true}, \text{false}\},$$

где $f_{\text{Конкр.}}$ – предикат конкретизации, определяющий принадлежность объектов классам (типам) “Обработка”, “Жетон” и “Узел”; $f_{\text{Агент}}$ – предикат, определяющий роль агента для запросов каждого из трех видов; $f_{\text{Разр.}}$ – предикат, определяющий “разрешающую” роль жетона; $f_{\text{Объект}}$ – предикат, определяющий роль объекта для каждого узла сети. Определены также функции для нумерации объектов (биекции): $f_{\text{Инд}_1}: T \rightarrow I$, $f_{\text{Инд}_2}: Y \rightarrow I$ и инъекция $f_{\text{Инд}_3}: P \rightarrow \omega$, где ω – множество натуральных чисел. Унарные предикаты $f_P: P \rightarrow \{\text{true}, \text{false}\}$, $f_Y: Y \rightarrow \{\text{true}, \text{false}\}$ и $f_T: T \rightarrow \{\text{true}, \text{false}\}$ определяют занятость объектов, принадлежащих множествам объектов P , Y и T соответственно. Для аналогичной цели используются и унарные предикаты $f_{S_1}: S_1 \rightarrow \{\text{true}, \text{false}\}$, $f_{S_2}: S_2 \rightarrow \{\text{true}, \text{false}\}$ и $f_E: E \rightarrow \{\text{true}, \text{false}\}$.

Следующие выражения для модулей СеАМ описывают выполнение предопределенного (s_1) и непредопределенного (s_2) запросов соответственно:

$$m_{s_1} = [f_{S_1}(s_1) \& (f_{\text{Запр.}}(s_1) \neq 0)] [(\exists!(t \in T, y \in Y)f_T(t) \& f_Y(y) \& \\ \& (f_{\text{Запр.}}(s_1) = f_{\text{Инд}_1}(t)) \& (f_{\text{Инд}_1}(t) = f_{\text{Инд}_2}(y)))] [(\exists!p \in P)f_P(p)] \\ (\{f_{\text{Агент}}(p, s_1) \leftarrow \text{true}, f_{\text{Разр.}}(p, t) \leftarrow \text{true}, f_{\text{Объект}}(p, y) \leftarrow \text{true},$$

$$\begin{aligned}
&f_p(p) \leftarrow \text{false}, f_T(t) \leftarrow \text{false}, f_Y(y) \leftarrow \text{false}, f_{s_1}(s_1) \leftarrow \text{false}, \\
&f_{\text{запр.}}(s_1) \leftarrow 0 \} \vee R^E) \vee R^E) \vee R^E); \\
&m_{s_2} = [f_{s_2}(s_2) \& (f_{\text{запр.}}(s_2) \neq 0)] ([\exists! (t \in T, y \in Y) f_T(t) \& f_Y(y) \& \\
&\& (f_{\text{Инд}_1}(t) = f_{\text{Инд}_2}(y))] ([\exists! p \in P) f_p(p)] \\
&(\{f_{\text{Агент}}(p, s_2) \leftarrow \text{true}, f_{\text{Разр.}}(p, t) \leftarrow \text{true}, f_{\text{Объект}}(p, y) \leftarrow \text{true}, \\
&f_p(p) \leftarrow \text{false}, f_T(t) \leftarrow \text{false}, f_Y(y) \leftarrow \text{false}, f_{s_2}(s_2) \leftarrow \text{false}, \\
&f_{\text{запр.}}(s_2) \leftarrow 0 \} \vee R^E) \vee R^E) \vee R^E).
\end{aligned}$$

В данных выражениях унарная функция $f_{\text{запр.}}$ задает номер запрошенной единицы ресурса R (одного из узлов сети ВЗУ); нулевое значение данной функции означает, что запрос выполнен.

Модуль m_{s_1} начинает выполняться, если запрос s_1 готов к выполнению ($f_{s_1}(s_1) = \text{true}$) и установлен его главный параметр – номер требуемой единицы ресурса R . Данный номер представлен ненулевым значением функции $f_{\text{запр.}}(s_1)$. В выражении, описывающем работу модуля m_{s_2} , также используется унарная функция $f_{\text{запр.}}$, но поскольку запрос $s_2 \in S_2$ требует любую свободную единицу ресурса R , учитываются лишь нулевое или ненулевое значения данной функции.

Предложены методы и средства проектирования, основанные на непосредственно исполняемых формальных спецификациях и обеспечивающие построение неограниченно масштабируемых систем с автоматизацией процессов реконфигурации структуры и миграции данных. Данные спецификации адекватно описывают процессы управления путями доступа к ресурсам, поиска и выборки данных, копирования и репликации данных.

Предложена функционально-структурная организация распределенной системы управления сетью хранения данных, являющаяся результатом развития ее концептуальной модели до системы исполняемых логико-алгебраических спецификаций, благодаря чему разработаны эффективные способы пространственно-временного распределения управляющих воздействий при организации приоритетного обслуживания запросов к сети хранения, рассматриваемой в качестве распределенного ресурса с многими единицами. Анализ приоритетных механизмов по построенной имитационной поведенческой модели показал, что от выбора приоритетной дисциплины, реализуемой в сетевой среде, в значительной степени зависит время обработки различных классов запросов к сети хранения и обработки данных.

Представлены исполняемые формальные спецификации для реализации сетевой машины баз данных. В том числе специфицировано выполнение операций над сегментированными отношениями в сетевой среде и выполнение операций реляционной алгебры в сетевой среде. Отличительной особенностью реализации является интерпретация операций сетями абстрактных машин, реализуемых виртуальными машинами в произвольной сетевой FS-среде, адекватной архитектуре большинства современных систем и сетей хранения и обработки данных. Структурные особенности построения

интеллектуальных ячеек процессора баз данных описаны в четвертой главе, а дополнительное описание сетевой реализации машины базы данных представлено в приложении 1.

В шестой главе отражены вопросы практического использования методов архитектурного моделирования, виртуализации и реализации многофункциональных систем и сетей внешних запоминающих устройств с параллельной организацией поиска и передачи данных. На конкретных примерах апробирована базирующаяся на данных методах технология для проектирования многофункциональных систем и сетей внешних запоминающих устройств. Отмечено, что положенная в основу данной технологии интерпретация сценарных сетей сетями абстрактных машин удобна при создании таких систем и сетей хранения и обработки данных, в которых стираются грани между сетевой операционной системой, распределенной системой управления базой данных и распределенным приложением.

Разработаны сценарные и логико-алгебраические модели систем ВЗУ с централизованной и децентрализованной сетевой архитектурой, с обычными и RAID-подобными режимами работы. На примере данных моделей продемонстрирована возможность трансформаций сетей абстрактных машин при декомпозиционно-композиционном подходе в архитектурном проектировании систем; при этом в качестве исходных используются сценарные представления.

Рассмотрены логико-алгебраические спецификации для операции барьерной синхронизации процессов, необходимой, например, в виртуальных реплицированных сетевых системах хранения данных и распределенных DRAID-системах (DRAID – distributed RAID). Предложены два варианта реализации барьерной синхронизации процессов. Первый вариант реализации основан на использовании формализма СеАМ. Ниже представлены логико-алгебраические выражения для каузально связанных модулей, участвующие в барьерной синхронизации (для агента-сервера a_i):

$$\begin{aligned} m_{ai} &= \dots f_{ai,1}(a_i) \leftarrow \text{true} \dots ; \\ m_{ai,1} &= [f_{ai,1}(a_i) \& (\exists! x \in X) f_s(x)] (\{f_s(x) \leftarrow \text{false}, f_{ai,1}(a_i) \leftarrow \text{false}, \\ & f_{ai,2}(a_i) \leftarrow \text{true}\} \vee R^E); \\ m_{ai,2} &= [f_{ai,2}(a_i) \& \neg(\exists x \in X) f_s(x)] (\{f_{ai,2}(a_i) \leftarrow \text{false}, f_{ai,3}(a_i) \leftarrow \text{true}\} \vee R^E); \\ m_{ai,3} &= [f_{ai,3}(a_i) \dots ; \\ x \in X, X &= \{x_1, x_2, \dots, x_n\}, i = 1, 2, \dots, n. \end{aligned}$$

Каждый из абстрактных агентов $a_1, a_2, \dots, a_n$ выполняет каузально связанные через соответствующие предикаты модули, выражения для которых построены на основе α -дизъюнкций и блоков обновления предикатов. Таким образом, каждому агенту $a_i, i = 1, 2, \dots, n$, соответствуют действия одноименного программного агента-сервера, интерпретирующего СеАМ-выражения для модулей. Так, агент-сервер a_i , “перемещаясь” по сети абстрактных машин, точнее, по ее узлам $m_{ai}, m_{ai,1}, m_{ai,2}, m_{ai,3}$, реализует процесс аб-

страктного агента a_i . Предикат $f_{ai,1}$ используется для организации каузальной связи: “по завершении работы модуль m_{ai} инициирует работу модуля $m_{ai,1}$ ”. Таким же образом реализуются каузальные связи между парами модулей $m_{ai,1}$ и $m_{ai,2}$, $m_{ai,2}$ и $m_{ai,3}$.

Второй пример демонстрирует один из вариантов реализации барьерной синхронизации модулями РСeAM. В этих реализациях модулей используется операция α -итерация и темпоральная операция “;” последовательного выполнения действий (упрощенное обозначение для операции “ $\uparrow$ ”) Агенты-серверы $a_1, a_2, \dots, a_n$, реализуя одноименные процессы, интерпретируют соответствующие выражения.

Логико-алгебраические выражения для модулей РСeAM имеют следующий вид:

$$\begin{aligned} m_{a1} &= (\dots [(\tilde{\exists}!x \in X) f_s^+(x)] \{R^E\}; f_s(x) \leftarrow \text{false}; f_s^-; [\neg(\exists x \in X) f_s^+(x)] \{R^E\}; f_s^-; \dots); \\ m_{a2} &= (\dots [(\tilde{\exists}!x \in X) f_s^+(x)] \{R^E\}; f_s(x) \leftarrow \text{false}; f_s^-; [\neg(\exists x \in X) f_s^+(x)] \{R^E\}; f_s^-; \dots); \\ &\dots \\ m_{ai} &= (\dots [(\tilde{\exists}!x \in X) f_s^+(x)] \{R^E\}; f_s(x) \leftarrow \text{false}; f_s^-; [\neg(\exists x \in X) f_s^+(x)] \{R^E\}; f_s^-; \dots); \\ &\dots \\ m_{an} &= (\dots [(\tilde{\exists}!x \in X) f_s^+(x)] \{R^E\}; f_s(x) \leftarrow \text{false}; f_s^-; [\neg(\exists x \in X) f_s^+(x)] \{R^E\}; f_s^-; \dots). \end{aligned}$$

Каждое выражение для операции α -итерации содержит формулу (в квадратных скобках) для условия, при ложности которого выполняется выражение в фигурных скобках (в данном случае тождественное правило R^E обновления). При истинности данного условия и после проверки этого факта агентом-сервером начинается выполнение выражения, следующего за выражением в фигурных скобках.

Выполнение α -итерации в выражениях РСeAM имеет свою специфику, связанную с блокировками функций и предикатов. В частности, при выполнении первой α -итерации в выражении для модуля m_{ai} осуществляется предварительная блокировка предиката f_s (этот факт отмечается символом “+”). Затем агент-сервер a_i проверяет возможность выбора одного из значений предметной переменной x . Если такой выбор сделан, то осуществляется выход из цикла и далее выбранное истинное значение предиката f_s заменяется на ложное, а затем данный предикат разблокируется (этот факт отмечается символом “-”). Если же выбор не удалось сделать, то предикат f_s разблокируется и агент-сервер a_i снова становится в очередь на захват данного предиката с последующей проверкой условия в квадратных скобках на истинность. В данной очереди уже могут находиться другие агенты-серверы, которые должны выполнить аналогичные действия над предикатом f_s .

Апробирована методология построения сложных имитационных поведенческих моделей на основе FS-технологии. В состав данных моделей включаются подмодели времязадающих устройств, систем синхронизации и

прерывания, определенные с помощью одного и того же математического аппарата исполняемых формальных спецификаций. Показано, что имитационные поведенческие модели данного класса также возможно использовать и для моделирования систем или сетей хранения и обработки данных как систем массового обслуживания для вероятностной оценки производительности при выборе того или иного варианта реализации общей логической структуры. В этом случае в имитационной модели используются подмодели сбора статистических данных о загрузке устройств, об очередях и времени ожидания каких-либо событий. Все подмодели определены формально с помощью логико-алгебраических представлений сетей абстрактных машин, что позволило понизить затраты на построение общей модели и последующую реализацию комплекса управляющих программ.

На рисунке 1 представлен пример общей схемы взаимодействий в одной из реализаций сети внешнего хранения и обработки данных. Сеть в целом реализует сервис DRAIDService, а непосредственное взаимодействие клиента с сетевыми дисками реализовано через драйвер взаимодействия DRAIDDriver системы виртуального дискового массива DRAID. Этот драйвер представляет собой интерфейсную часть реализации системы и предназначен для обмена информацией с драйверами сетевых дисков. На стороне дискового массива непосредственными участниками взаимодействий являются драйверы сетевых дисков NetDiskDriver, взаимодействующие по специальному протоколу. Большинство взаимодействий между компонентами осуществляется через FS-пространство с помощью агентов-серверов, реализующих логико-алгебраические спецификации СеАМ. Данная сеть предназначена для организации на ее основе разнообразных систем хранения и обработки данных, например, распределенных хранилищ, использующих фрагментацию данных, и виртуальных RAID-массивов.

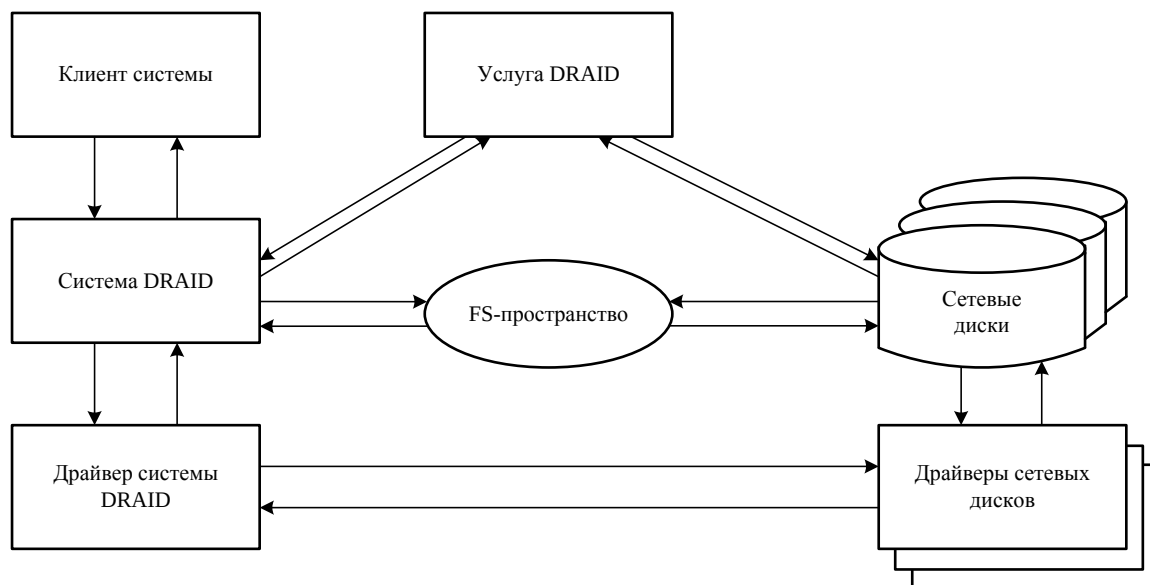


Рисунок 1 – Схема взаимодействия компонент в сети внешнего хранения и обработки данных DRAID

FS-технология хорошо интегрируется с известными сетевыми программными платформами, например, с платформой Jini, предложенной фирмой Sun Microsystems. Программная платформа Jini позволяет связывать сетевые устройства в единую систему независимо от особенностей их операционных систем (ОС) и интерфейсов. Например, некоторые услуги и сервисы технологии Jini использованы в одной из сетевых реализаций системы внешнего хранения данных со схемой взаимодействия клиента с устройствами внешней памяти (ВП), приведенной на рисунке 2.

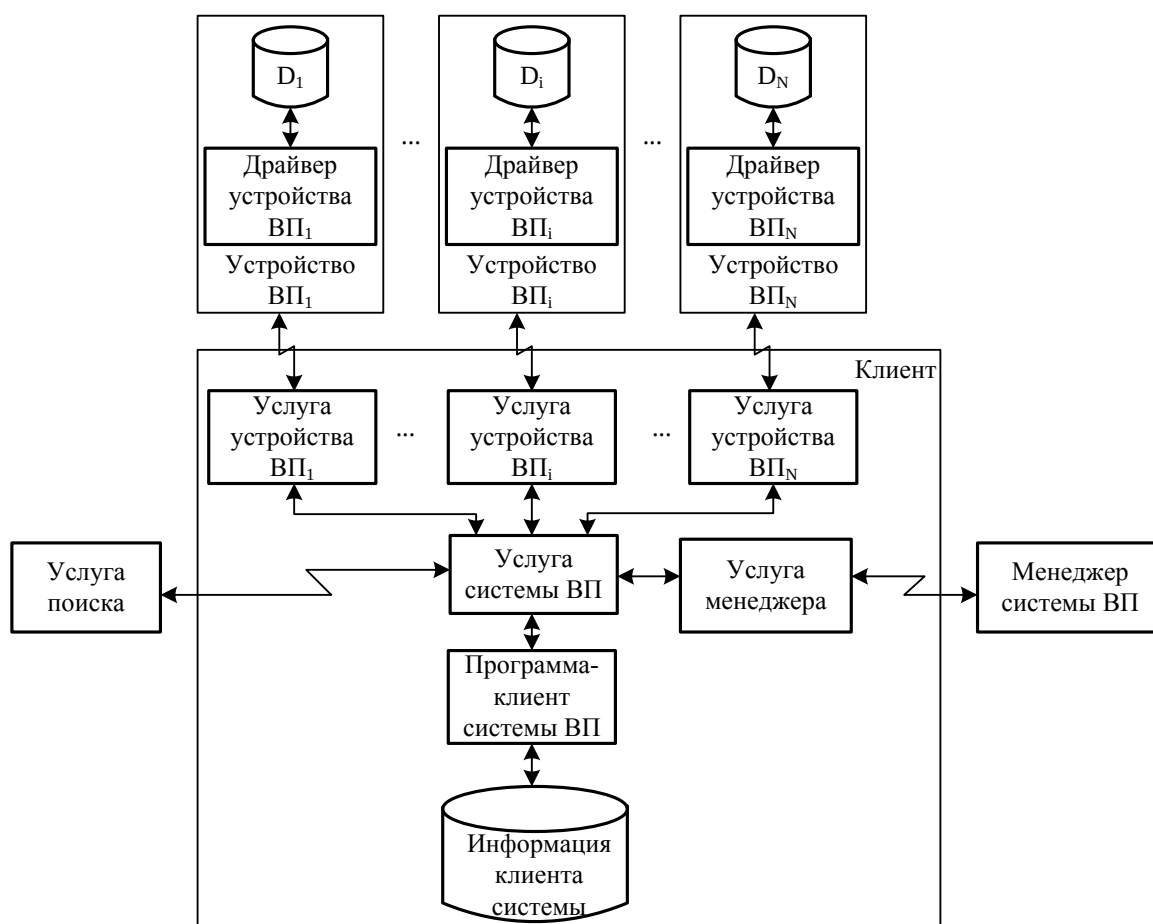


Рисунок 2 – Схема взаимодействия клиента с устройствами сетевой внешней памяти

Основные уровни, на которых происходит взаимодействие компонент в сети внешнего хранения и обработки данных при реализации данной схемы взаимодействия, показаны на рисунке 3. Менеджер системы ВП и услуга поиска Jini представляют серверную сторону. Соответственно, на клиентской машине пользователя реализованы такие же уровни взаимодействия. Между клиентами системы, услугой поиска и менеджером системы ВП взаимодействие ведется на уровне протоколов поиска, обнаружения и присоединения услуг Jini. Использована также RMI-технология построения распределенных приложений, предложенная в спецификации языка Java. Исследованы также вопросы интеграции FS-технологии с другими сетевыми технологиями – CORBA, MPI и технологиями “клиент-сервер”.

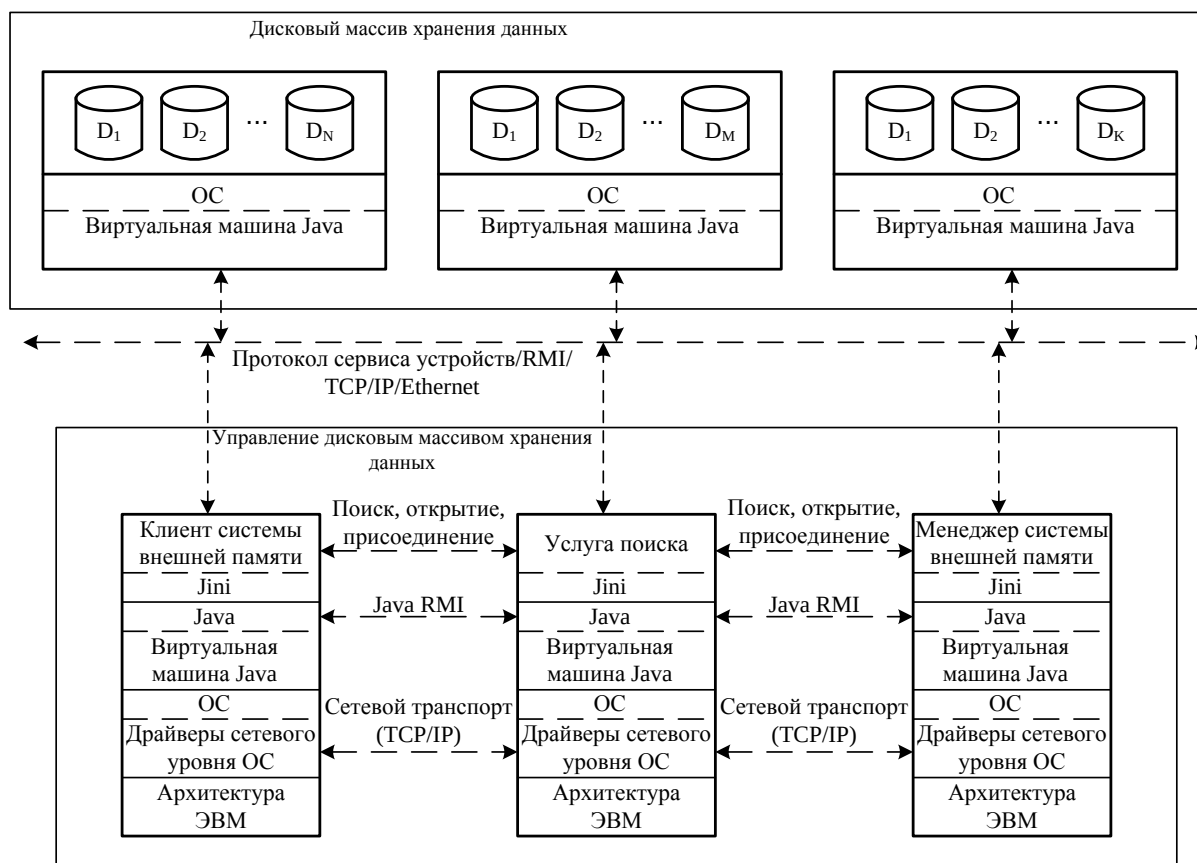


Рисунок 3 – Архитектура сети внешнего хранения и обработки данных

Проведены серии имитационных экспериментов, подтвердивших работоспособность разработанных на основе новых технологий имитационных поведенческих моделей; получены основные характеристики производительности сети внешних запоминающих устройств, показавшие, что за счет функционально-структурной реорганизации может быть в полтора-два раза повышена производительность систем, а за счет распараллеливания операций в сети хранения обеспечивается линейный рост производительности, замедляемый вследствие затрат на коммуникации между устройствами и на передачу данных по сети. На основе разработанных моделей отлажены исполняемые формальные спецификации, реализованные в виде комплекса управляющих программ для сети ВЗУ с параллельным выполнением операций и расслоением записей. Проведено экспериментальное исследование сети хранения данных при учете различного расположения точек виртуализации, которое позволило оценить качество предложенных архитектурных решений и пригодность к реализации на их основе сетевых хранилищ с параллельной организацией доступа к данным.

Шесть приложений к основному тексту диссертации содержат описание реализации и функционирования процессора базы данных в сетевой среде (**приложение 1**); описание функционально-структурной организации многофункционального узла сети внешних запоминающих устройств,

включающего кэш-память (**приложение 2**); описание функционально-структурной организации многофункциональной системы внешних запоминающих устройств, подключаемой к вычислительной сети (**приложение 3**); описание функционально-структурной организации системы внешних запоминающих устройств на дисках, включающей машину базы данных (**приложение 4**); описание сетевой реализации структурированной виртуальной памяти, или FS-пространства (**приложение 5**) документы, подтверждающие внедрение и использование результатов диссертации (**приложение 6**).

ОСНОВНЫЕ РЕЗУЛЬТАТЫ РАБОТЫ

Основным результатом диссертационной работы является создание теоретических основ, методов и средств архитектурного моделирования и проектирования, функционально-структурной реализации и виртуализации систем и сетей внешнего хранения и обработки данных, обладающих расширенными функциональными возможностями и высокой производительностью. При проведении теоретических и практических исследований по тематике диссертации получены следующие научные и практические результаты.

1. Предложен метод архитектурного моделирования и проектирования проблемно-ориентированных систем и сетей хранения и обработки данных, эффективность которого обеспечивается согласованностью исходной концептуальной модели с формально определенными в процессе иерархического проектирования имитационными поведенческими моделями. В отличие от известных методов указанная согласованность достигается путем использования определенных в работе сценарных представлений и логико-алгебраических описаний, обеспечивающих большую адекватность моделей предметной области.

2. Дано теоретическое обоснование на основе определенных в диссертационной работе эволюционирующих иерархических алгебраических систем, иерархических сетей абстрактных машин основного класса систем согласования и координации процессов и объектов как базовой платформы для построения прототипного и рабочего программного и аппаратного обеспечения систем и сетей внешнего хранения и обработки данных, обеспечивающей более адекватное отображение сети алгоритмических модулей на физическую сеть при реализации структурных способов повышения производительности. Использование данной платформы позволяет уменьшить последствия семантического разрыва между инфраструктурой систем и сетей хранения данных и алгоритмическими структурами, обеспечивающими их многофункциональность.

3. Предложен метод “активной классной доски”, послуживший основой для внешней и внутренней интеллектуализации систем и сетей внешнего хранения и обработки данных, эффективность реализации которого, в отличие от известных, обеспечивается внедрением механизмов, основанных на

формальных исполняемых спецификациях и на декларативно-процедурной структуризации представления знаний о предметной области.

4. Дана концепция развития архитектуры, отличной от фон-неймановской на системном уровне реализации систем и сетей хранения и обработки данных и способствующая развитию принципов прозрачности в управлении ресурсами и масштабируемости, организации эффективного множественного доступа к средствам внешнего хранения и обработки структурированных и неструктурированных данных. Такая концепция отличается использованием абстракции общего структурированного пространства информационных объектов, посредством которого устанавливаются каузальные связи между модулями сети абстрактных машин, а функционирование системы и ее внутренние структурно-логические связи отображаются в модификациях предикатов и функций абстрактного пространства.

5. Предложены новые принципы функционально-структурной организации на основе формального описания концепции согласования и координации процессов и объектов. Данные принципы положены в основу пространственно-временной организации управляющих воздействий в системах и сетях внешнего хранения и обработки данных и обеспечивают согласованные взаимодействия компонент. Использование указанных принципов позволяет осуществлять эффективную реализацию основных методов управления в функционально-централизованных, функционально-децентрализованных и топологически-распределенных системах.

6. Разработан метод интерпретации пространственных и временных сценариев функционирования систем и сетей хранения и обработки данных сетями абстрактных машин, который отличается возможностью эволюционного перехода от концептуальных моделей к непосредственно исполняемым формальным логико-алгебраическим спецификациям посредством представления каждого концептуального графа модулем сети абстрактных машин, что снижает трудоемкость реализации комплекса аппаратно-программных средств систем и сетей хранения и обработки данных.

7. Предложен метод расширения функциональных возможностей систем и сетей внешнего хранения и обработки данных за счет включения дополнительных функций обработки данных в узлах (точках) виртуализации. Предлагаемая концепция отличается реализацией на основе произвольной архитектуры вычислительных средств, что в большей степени расширяет функциональные возможности виртуализированной архитектуры и сокращает затраты на ее реализацию.

8. Разработаны концептуальные, имитационные поведенческие модели систем и сетей хранения и обработки данных, сетевая среда для реализации получаемых формальных спецификаций, структура и реализация системы синтеза структур и алгоритмического обеспечения систем и сетей внешнего хранения и обработки данных, позволившие осуществить эффективную архитектурную и

функционально-структурную организацию данных систем и сетей с развитыми функциональными возможностями и высокой производительностью.

9. Предложена архитектурная и функционально-структурная организация систем и сетей внешнего хранения и обработки данных с развитыми функциональными возможностями и высокой производительностью, ориентированная на широкий спектр универсальных и специализированных приложений, в том числе:

- многофункционального узла сети внешних запоминающих устройств, включающего кэш-память;
- многофункциональной системы внешних запоминающих устройств, подключаемой к вычислительной сети;
- системы внешних запоминающих устройств, включающей машину баз данных; в основу построения трех данных систем положены новые технические решения, отличающиеся от известных возможностью распараллеливания операций поиска и обмена данными и обеспечивающие повышение производительности, что подтверждается авторским свидетельством;
- процессора базы данных, реализуемого в сетевой среде;
- сетевой среды для реализации виртуализированных архитектур систем и сетей внешнего хранения и обработки данных.

Функционально-структурная организация данных систем и сетей отличается структурной интерпретацией в конкретных программно-аппаратных средах концептуальных и формально определенных имитационных поведенческих моделей, обеспечивающей в конечном итоге эффективность интеграции и консолидации вычислительных ресурсов с ресурсами хранения данных и системой коммуникаций.

ОСНОВНЫЕ ПУБЛИКАЦИИ ПО ТЕМЕ ДИССЕРТАЦИИ

Статьи в изданиях, рекомендованных ВАК РФ

1. Зинкин, С. А. Иерархические сети абстрактных машин и виртуализация интеллектуальных систем внешнего хранения и обработки данных / С. А. Зинкин // Известия высших учебных заведений. Поволжский регион. Технические науки. – 2009. – № 2. – С. 25–38.

2. Зинкин, С. А. Согласование и координация объектов и процессов в агентно-ориентированных системах и сетях хранения данных / С. А. Зинкин // Вопросы радиоэлектроники. – 2009. – № 4. – С. 83–96.

3. Зинкин, С. А. Реализация барьерной синхронизации и управление процессами в виртуальном сетевом дисковом массиве / С. А. Зинкин // Информационные технологии. – 2008. – № 12. – С. 22–29.

4. Зинкин, С. А. Элементы новой объектно-ориентированной технологии для моделирования и реализации систем и сетей хранения и обработки данных / С. А. Зинкин // Информационные технологии. – 2008. – № 10. – С. 20–27.

5. Зинкин, С. А. Элементы технологии иерархического концептуального моделирования и реализации систем и сетей хранения и обработки данных / С. А. Зинкин // Известия высших учебных заведений. Поволжский регион. Технические науки. – 2008. – № 4. – С. 3–15.

6. Зинкин, С. А. Управление распределенными ресурсами в сети внешних запоминающих устройств / С. А. Зинкин // Вопросы радиоэлектроники. – 2008. – № 5. – С. 85–97.

7. Зинкин, С. А. Разработка интеллектуальных систем управления внешней памятью ЭВМ и сетей с расширенными функциональными возможностями / С. А. Зинкин // Известия высших учебных заведений. Поволжский регион. Технические науки. – 2008. – № 2. – С. 3–15.

8. Зинкин, С. А. Концептуальное моделирование многофункциональных систем и сетей внешних запоминающих устройств / С. А. Зинкин // Вопросы радиоэлектроники. – 2008. – № 5. – С. 76–85.

9. Зинкин, С. А. Функционально-структурная реализация интеллектуальных систем управления внешней памятью ЭВМ и сетей / С. А. Зинкин // Известия высших учебных заведений. Поволжский регион. Технические науки. – 2008. – № 1. – С. 14–21.

10. Зинкин, С. А. Сети абстрактных машин высших порядков в проектировании систем и сетей хранения и обработки данных (механизмы интерпретации и варианты использования) / С. А. Зинкин // Известия высших учебных заведений. Поволжский регион. Технические науки. – 2007. – № 4. – С. 37–51.

11. Зинкин, С. А. Сети абстрактных машин высших порядков в проектировании систем и сетей хранения и обработки данных (базовый формализм и его расширения) / С. А. Зинкин // Известия высших учебных заведений. Поволжский регион. Технические науки. – 2007. – № 3. – С. 13–22.

12. Зинкин, С. А. Самомодифицируемые сценарные модели функционирования систем и сетей хранения и обработки данных (реализация и свойства сценарных моделей) / С. А. Зинкин // Известия высших учебных заведений. Поволжский регион. Технические науки. – 2007. – № 2. – С. 13–21.

13. Зинкин, С. А. Самомодифицируемые сценарные модели функционирования систем и сетей хранения и обработки данных (базовый формализм и темпоральные операции) / С. А. Зинкин // Известия высших учебных заведений. Поволжский регион. Технические науки. – 2007. – № 1. – С. 3–12.

14. Вашкевич, Н. П. Применение методологии и принципов нечеткой логики в информационной системе «Электронная история болезни» / Н. П. Вашкевич, С. А. Зинкин, М. М. Макаров // Вопросы радиоэлектроники. – 2007. – № 2. – С. 5–14.

15. Зинкин, С. А. Алгебра сценариев для спецификации операционной семантики активных сетей хранения и обработки данных / С. А. Зинкин //

Известия высших учебных заведений. Поволжский регион. Технические науки. – 2004. – № 2. – С. 96–107.

Публикации в других изданиях

16. Зинкин, С. А. Сетевая FS-архитектура машин баз данных и исполняемые формальные спецификации / С. А. Зинкин // Вестник Пермского государственного университета. Научный журнал. Серия “Математика. Механика. Информатика”. – 2008. – № 4(20). – С. 169–181.

17. Коллективное поведение агентов в сетях хранения и обработки данных / Н. П. Вашкевич, С. А. Зинкин, В. А. Киреев, А. В. Прошкин // Приложение к журналу “Информационные технологии”. – 2003. – № 9. – 24 с.

18. Зинкин, С. А. Мобильные агенты в TCP/IP сетях: особенности технологии и проблемы безопасности / С. А. Зинкин, В. А. Киреев, А. В. Прошкин // Телекоммуникации. – 2002. – № 9. – С. 30–39.

19. Вашкевич, Н. П. Использование методов искусственного интеллекта при проектировании интеллектуальных систем управления внешней и массовой памятью ЭВМ / Н. П. Вашкевич, С. А. Зинкин, В. Н. Дубинин // Известия АН СССР. Техническая кибернетика. – 1991. – № 5. – С. 63–74.

20. Вашкевич, Н. П. Структурный подход к проектированию мультипроцессорной вычислительной системы управления базой данных / Н. П. Вашкевич, С. А. Зинкин, В. П. Кулагин // Известия высших учебных заведений. Приборостроение. – 1983. – Т. XXVI. – № 9. – С. 15–24.

21. Зинкин, С. А. Сети абстрактных машин и концептуальное моделирование многофункциональных систем и сетей внешних запоминающих устройств / С. А. Зинкин // Перспективные технологии искусственного интеллекта: сб. тр. Междунар. науч.-практ. конф. – Пенза : Информационно-издательский центр ПензГУ. – 2008. – С. 253–260.

22. Зинкин, С. А. Внешняя и внутренняя интеллектуализация систем и сетей хранения и обработки данных / С. А. Зинкин // Новые информационные технологии и системы: тр. VIII Междунар. науч.-техн. конф. Часть 1. – Пенза : Изд-во Пенз. гос. ун-та. – 2008. – С. 29–50.

23. Зинкин, С. А. Методы разработки систем и сетей хранения и обработки данных на основе иерархии концептуальных поведенческих моделей / С. А. Зинкин // Новые информационные технологии и системы: тр. VIII Междунар. науч.-техн. конф. Часть 1. – Пенза : Изд-во Пенз. гос. ун-та. – 2008. – С. 17–29.

24. Зинкин, С. А. Сценарные модели функционирования систем и сетей хранения и обработки данных / С. А. Зинкин // Новые информационные технологии и системы: тр. VII Междунар. науч.-техн. конф. Часть 1. – Пенза : Изд-во Пенз. гос. ун-та. – 2006. – С. 117–123.

25. Зинкин, С. А. Интеграция сетевых и информационных технологий на основе парадигм искусственного интеллекта / С. А. Зинкин // Новые информационные технологии и системы: тр. VII Междунар. науч.-техн. конф. Часть 1. – Пенза : Изд-во Пенз. гос. ун-та. – 2006. – С. 108–117.
26. Зинкин, С. А. Интеграция сетевых, информационных и кластерных технологий в TCP/IP сетях. Метод. указания / С. А. Зинкин, В. В. Швецов, Е. А. Шумилкина. – Пенза : Изд-во Пенз. гос. ун-та. – 2006. – 76 с.
27. Зинкин, С. А. Организация параллельного дедуктивного вывода в распределенной среде / С. А. Зинкин, В. В. Швецов // Математическое и программное обеспечение вычислительных систем: межвуз. сб. науч. тр. – Рязань: Рязан. гос. радиотехн. акад. – 2005. – С. 74–78.
28. Зинкин, С. А. Реализация барьерной синхронизации и управление процессами в FS-пространстве / С. А. Зинкин // Новые информационные технологии и системы: тр. VI Междунар. науч.-техн. конф. Часть 1. – Пенза : Изд-во Пенз. гос. ун-та. – 2004. – С. 217–226.
29. Волчихин, В. И. Новые парадигмы сетевых информационных технологий / В. И. Волчихин, Н. П. Вашкевич, С. А. Зинкин // Новые информационные технологии и системы: тр. VI Междунар. науч.-техн. конф. Часть 1. – Пенза : Изд-во Пенз. гос. ун-та. – 2004. – С. 168–182.
30. Зинкин, С. А. Эволюции сетей хранения и обработки данных / С. А. Зинкин // Новые информационные технологии и системы: тр. VI Междунар. науч.-техн. конф. Часть 1. – Пенза : Изд-во Пенз. гос. ун-та. – 2004. – С. 192–200.
31. Зинкин, С. А. Концептуальная схема реализации сетевой информационной технологии FUNCTION SPACES / С. А. Зинкин // Новые информационные технологии и системы: тр. VI Междунар. науч.-техн. конф. Часть 1. – Пенза : Изд-во Пенз. гос. ун-та. – 2004. – С. 208–217.
32. Зинкин, С. А. Виртуальные топологии сетей хранения и обработки данных / С. А. Зинкин // Новые информационные технологии и системы: тр. VI Междунар. науч.-техн. конф. Часть 1. – Пенза : Изд-во Пенз. гос. ун-та. – 2004. – С. 200–208.
33. Зинкин, С. А. Использование интерфейса передачи сообщений для организации доступа к распределенным данным / С. А. Зинкин, В. В. Швецов // Проблемы передачи и обработки информации в сетях и системах телекоммуникаций: материалы 13-й Междунар. науч.-техн. конф. – Рязань : Рязан. гос. радиотехн. академия. – 2004. – С. 156–158.
34. Зинкин, С. А. Алгебраические методы спецификации операционной семантики сетей хранения и обработки данных с изменяемой топологией / С. А. Зинкин // Труды Международного юбилейного симпозиума «Актуальные проблемы науки и образования». Т. 2. – Пенза : Информационно-издательский центр ПГУ, 2003. – С. 409–418.

35. Зинкин, С. А. Мультиагентные системы и архитектура M2PM2D сетей хранения и обработки данных: макроструктурная теория и практическая реализация / С. А. Зинкин // Труды Международного юбилейного симпозиума «Актуальные проблемы науки и образования». Т. 2. – Пенза : Информационно-издательский центр ПГУ, 2003. – С. 419–424.
36. Зинкин, С. А. Семантические сети в управлении мобильными агентно-ориентированными вычислениями / С. А. Зинкин // Новые информационные технологии и системы: Материалы V Международной НТК. – Пенза, 2002. – С. 48–52.
37. Зинкин, С. А. A SAN/NAS-like platform for the intelligent distributed systems design / С. А. Зинкин // Новые информационные технологии и системы: Материалы V Междунар. НТК. – Пенза, 2002. – С. 56–60.
38. Зинкин, С. А. Мобильные агентно-базированные вычисления и семантические сети / С. А. Зинкин // Материалы XIII Международной школы-семинара “Синтез и сложность управляющих систем”. Ч. I, Ч. II. – М. : МГУ, 2002. – С. 89–94.
39. Зинкин, С. А. Макроструктурная теория распределенных мобильных систем хранения и обработки данных / С. А. Зинкин // Материалы XIII Международной школы-семинара “Синтез и сложность управляющих систем”. Ч. I. Ч. II. – М. : МГУ, 2002. – С. 84–89.
40. Зинкин, С. А. Анализ состояния распределенной системы хранения данных / С. А. Зинкин // Межвузовский сборник научных трудов. – Пенза : Изд-во Пенз. гос. ун-та, 2002. – С. 127–137.
41. Зинкин, С. А. Вероятностные модели поиска данных в вычислительных сетях / С. А. Зинкин // Материалы XII Международной школы-семинара “Синтез и сложность управляющих систем”. Ч. I, Ч. II. – М. : МГУ, 2001. – С. 90–94.
42. Зинкин, С. А. Распределенная модель хранения данных для корпоративных сетей / С. А. Зинкин // Новые информационные технологии и системы: материалы IV Междунар. НТК. – Пенза, 2000. – С. 102–103.
43. Зинкин, С. А. Организация управления распределенным хранилищем данных / С. А. Зинкин // Новые информационные технологии и системы: материалы IV Междунар. НТК. – Пенза, 2000. – С. 100–102.
44. Зинкин, С. А. Метод производящих функций в исследовании функционирования распределенной системы хранения данных / С. А. Зинкин // Новые информационные технологии и системы: материалы IV Междунар. НТК. – Пенза, 2000. – С. 103–106.
45. Сетевой агентно-ориентированный метакомпьютер / Н. П. Вашкевич, С. А. Зинкин, В. А. Киреев, А. В. Прошкин // Новые информационные технологии и системы: материалы IV Междунар. конф. – Пенза, 2000. – С. 106–109.

46. Зинкин, С. А. Анализ состояния распределенной системы хранения данных / С. А. Зинкин // Новые информационные технологии и системы: материалы IV Междунар. НТК.– Пенза, 2000. – С. 99–100.
47. Зинкин, С. А. Управление информационными потоками в распределенных системах памяти с использованием технологий мобильных агентов / С. А. Зинкин // Распознавание-99: сб. материалов IV Междунар. конф. – Курск, КГТУ, 1999. – С. 124–125.
48. Зинкин, С. А. Организация распределенной памяти для высокопроизводительных структур на платформе JINI и JAVASPACEs / С. А. Зинкин // Распознавание-99 : материалы IV Междунар. конф. – Курск, 1999. – С. 122–124.
49. Зинкин, С. А. Distributed evolving algebras meet Java aglets: executable specifications of virtual mobile metacomputing on the Internet platform / С. А. Зинкин // Новые информационные технологии и системы: материалы III Междунар. НТК.– Пенза, 1998. – С. 50–52.
50. Зинкин, С. А. Концепция мобильности в распределенных системах обработки информации / С. А. Зинкин // Материалы VII Всероссийского семинара “Нейроинформатика и её приложения”. – Красноярск, 1999. – С. 58.
51. Zinkin, S. A. The intellectual mass memory design for building high-performance computer systems and networks / S. A. Zinkin // Proceedings East-West International Conference “Information Technology in Design” (EWITD’94). Moscow, Russia, 5–9 September, Part 2. 1994. – P. 151–153.
52. Zinkin, S. A. Systolic nets modeling with simulation programs of different types / S. A. Zinkin // Proceedings East-West International Conference “Information Technology in Design” (EWITD’96). Moscow, Russia, 1–5 July. – P. 91–93.
53. Zinkin, S. A. The software and hardware architecture of a knowledge representation system / S. A. Zinkin // Proceedings East-West International Conference “Information Technology in Design” EWITD’96. – Moscow, Russia, 1–5 July. – P. 87–90.
54. Зинкин, С. А. Интеллектуальная дисковая память для построения систем обработки знаний / С. А. Зинкин // Сборник аннотаций IV Международной конференции “Дисковые запоминающие устройства” (ДЗУ’90). – Костенец, Болгария. – 1990. – С. 50–51.
55. Зинкин, С. А. Моделирование интеллектуальных систем массовой памяти / С. А. Зинкин // Сборник аннотаций IV Международной конференции “Дисковые запоминающие устройства” (ДЗУ’90). – Костенец, Болгария. – 1990. – С. 52–53.
56. А.с. 1462411 СССР. Система внешних запоминающих устройств на магнитных дисках/Н. П. Вашкевич, В. Н. Дубинин, А. М. Дудкин, С. Л. Зверев, С. А. Зинкин, Б. М. Раков. – № 3973458/24-10.–Опубл. 28.02.89. Бюл. № 8.
57. Зинкин, С. А. Проектирование функционально-децентрализованных вычислительных систем / С. А. Зинкин // Материалы докладов Второго ре-

гионального семинара “Распределенная обработка информации”. – Новосибирск : Изд-во СО АН СССР. – 1987. – С. 48.

58. Зинкин, С. А. Система моделирования функционально-децентрализованных систем на основе многоуровневых однородных сред / С. А. Зинкин // Материалы докладов Второго регионального семинара “Распределенная обработка информации”. – Новосибирск : Изд-во СО АН СССР. – 1987. – С. 66.

Зинкин Сергей Александрович

Развитие теоретических основ и методов
функционально-структурной организации систем
и сетей внешнего хранения и обработки данных

Специальность 05.13.15 – Вычислительные машины и системы

Специальность 05.13.13 – Телекоммуникационные системы
и компьютерные сети

Редактор *Т. В. Веденеева*

Технический редактор *Н. А. Вьялкова*

Корректор *Ж. А. Лубенцова*

Компьютерная верстка *Р. Б. Бердниковой*

Сдано в производство 29.09.2009. Формат 60x84¹/16.
Бумага офсетная. Печать офсетная. Усл. печ. л. 2,56.
Заказ № 446. Тираж 100.

Издательство Пензенского государственного университета.
440026, Пенза, Красная, 40.