

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ
ФЕДЕРАЦИИ**

ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ**



ПОБЕДИТЕЛЬ КОНКУРСА ИННОВАЦИОННЫХ ОБРАЗОВАТЕЛЬНЫХ ПРОГРАММ ВУЗОВ

НАУЧНО-ТЕХНИЧЕСКИЙ ВЕСТНИК

Выпуск 45

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ



САНКТ-ПЕТЕРБУРГ

2007

Выпуск содержит материалы XXXVI научной и учебно-методической конференции СПбГУ ИТМО «Достижения ученых, аспирантов и студентов университета в науке и образовании». Конференция была проведена 30 января - 2 февраля 2007 года Санкт-Петербургским государственным университетом информационных технологий, механики и оптики в сотрудничестве с

- ФГУП ВНЦ «ГОИ им. С.И. Вавилова»
- ФГУП НПК «ГОИ им. С.И. Вавилова»
- ФГУП НИТИОМ «ГОИ им. С.И. Вавилова»
- Институтом аналитического приборостроения РАН (ИАНП РАН)
- Институтом проблем машиноведения РАН (ИПМаш РАН)
- Комитетом по науке и высшей школе Санкт-Петербурга (КНВШ)
- ФГУП СПб ОКБ «Электроавтоматика»
- ВНИИМ им. Д.И. Менделеева
- ОАО «ЛОМО»
- ОАО «Техприбор»
- ЦНИИ «Электроприбор»
- ЗАО «БИ ПИТРОН»

В выпуске представлены работы, поддержанные финансированием в рамках:

- Федеральной целевой научно-технической программы «Исследования и разработки по приоритетным направлениям развития науки и техники на 2002–2006 гг» (Федеральное агентство по науке и инновациям);
 - аналитической ведомственной целевой программы «Развитие научного потенциала высшей школы (2006–08 гг.) (Федеральное агентство по образованию);
 - Федеральной целевой программы развития образования на 2006–2010 гг. (Федеральное агентство по образованию),
 - Российского фонда фундаментальных исследований,
- а также инициативные разработки.

Программный комитет конференции:

Васильев В.Н. (СПбГУ ИТМО) – председатель

Аронов А.М. (ОАО «ЛОМО»)	Маслов Ю.В. (ОАО «Техприбор»)
Викторов А.Д. (КНВШ)	Максимов А.С. (КНВШ)
Гатчин Ю.А. (СПбГУ ИТМО)	Медунецкий В.М. (СПбГУ ИТМО)
Дукельский К.В. (ФГУП ВНЦ «ГОИ им. С.И. Вавилова»)	Немолочнов О.Ф. (СПбГУ ИТМО)
Жигулин Г.П. – начальник ИКВО	Никифоров В.О. (СПбГУ ИТМО)
Иванов А.Ю. (СПбГУ ИТМО)	Парамонов П.П. (ФГУП СПб ОКБ «Электроавтоматика»)
Иванов А.В. (СПбГУ ИТМО)	Парфенов В.Г. (СПбГУ ИТМО)
Карасев В.Б. (ФГУП ВНЦ «ГОИ им. С.И. Вавилова»)	Пешехонов В.Г. (ЦНИИ «Электроприбор»)
Козлов С.А. (СПбГУ ИТМО)	Семенов А.Н. (СПбГУ ИТМО)
Колесников Ю.Л. (СПбГУ ИТМО)	Смирнов С.Б. (СПбГУ ИТМО)
Курочкин В.Е. (ИАНП РАН)	Стафеев С.К. (СПбГУ ИТМО)
Латыев С.М. (СПбГУ ИТМО)	Ткалич В.Л. (СПбГУ ИТМО)
Лукьянов Г.Н. (СПбГУ ИТМО)	Ханов Н.И. (ВНИИМ им. Д.И. Менделеева)
	Шехонин А.А. (СПбГУ ИТМО)
	Яблочников Е.И. (СПбГУ ИТМО)

Организационный комитет конференции:

Никифоров В.О. – председатель, Студеникин Л.М. – зам. председателя,
Казар Л.Н. – ученый секретарь, Горкина Н.М. Гусарова Н.Ф. Савельева Л.П.

ISSN 1819-222X

© Санкт-Петербургский государственный университет
информационных технологий, механики и оптики, 2007

**МЕТОДИКА ПОДГОТОВКИ БУДУЩИХ ИНЖЕНЕРОВ
СПЕЦИАЛЬНОСТИ «ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ
В ОБРАЗОВАНИИ» К ПРОЕКТНОЙ ДЕЯТЕЛЬНОСТИ****М.И. Потеев, М.В. Хлопотов**

Предлагается методика, построенная с использованием сквозной проектной подготовки и деловых игр в области проектной деятельности. Анализируются критерии качества цифровых информационных образовательных ресурсов.

Согласно Государственному образовательному стандарту высшего профессионального образования, выпускники специальности «Информационные технологии в образовании» получают квалификацию «инженер» и должны быть готовы к выполнению проектной, технологической, организационно-управленческой, научно-исследовательской и эксплуатационной деятельности.

Остановимся на проектной деятельности и рассмотрим некоторые составляющие методики подготовки к ней будущих инженеров указанной специальности. Для них объектом проектирования являются *цифровые информационные образовательные ресурсы (ЦИОР)* различных уровней сложности, а основным средством проектирования – информационные технологии. ЦИОР представляют собой образовательные ресурсы, представленные на электронных носителях в цифровой форме. Примерами ЦИОР являются: сайты университетов и библиотек, компьютерные обучающие и тестирующие программы, электронные учебники и учебно-методические комплексы.

Одним из подходов к формированию у будущих инженеров умения проектировать соответствующие объекты является использование проектного метода обучения [1]. По своей педагогической направленности, метод представляет собой деловую игру в области проектирования, состоящую в том, что специально сформированные микрогруппы студентов (в составе 3-5 человек) выполняют проектные задания преподавателя, распределив роли и координируя свою деятельность по достижению поставленной перед ними целью проектирования. После выполнения заданий всеми микрогруппами проводится заслушивание докладов их представителей и «защита» результатов работы перед членами всей студенческой группы.

Определим перечень компетенций, которые связаны с проектной деятельностью и должны быть сформированы у будущего инженера. Отметим основные педагогические особенности деловых игр в области проектирования. Покажем высокую эффективность рассматриваемого подхода для диагностики сформированных умений. Опишем опыт проведения деловых игр в области проектирования при подготовке будущих инженеров специальности «Информационные технологии в образовании».

Согласно подходу к классификации компетентностей, предлагаемому Н. Ф. Радионой, А. П. Тряпицыной и др., в структуре профессиональной компетентности будущего специалиста можно выделить ключевые, базовые и специальные компетентности [2].

Ключевые компетентности необходимы для любой профессиональной деятельности, связаны с успехом личности в быстро меняющемся мире и проявляются, прежде всего, в способности решать профессиональные задачи на основе использования информации, коммуникации, в том числе на иностранном языке, социально-правовых основ поведения личности в гражданском обществе.

Базовые компетентности отражают специфику определенной профессиональной деятельности (педагогической, медицинской, инженерной и т.д.).

Специальные компетентности отражают специфику конкретной предметной сферы профессиональной деятельности. Специальные компетентности рассматриваются как реализация ключевых и базовых компетентностей в области учебного предмета, конкретной области профессиональной деятельности.

Причем все три вида компетентности взаимосвязаны и развиваются одновременно. Это создает целостный образ специалиста и в конечном итоге обеспечивает становление его профессиональной компетентности как определенной целостности, как интегративной личностной характеристики.

Ключевые, базовые и специальные компетентности, взаимодействуя друг с другом, проявляются в процессе решения *профессиональных задач* разного уровня сложности в разных контекстах, с использованием определенного образовательного пространства [3].

Согласно Государственному образовательному стандарту, инженер любой специальности должен быть подготовлен к решению следующих типов профессиональных задач по виду проектно-конструкторской деятельности:

- определение целей проектирования, критериев эффективности, ограничений применимости;
- системный анализ объекта проектирования, предметной области, их взаимосвязей;
- выбор исходных данных для проектирования;
- разработка обобщенных вариантов решения проблемы, анализ этих вариантов, прогнозирование последствий, нахождение компромиссных решений в условиях многокритериальности и неопределенности, планирование реализации проекта;
- оценка надежности и качества функционирования объекта проектирования;
- разработка, согласование и выпуск всех видов проектной документации.

Воспользуемся рекомендациями В. С. Леднева [6], согласно которым в содержательной структуре обучения выделяют: *сквозные линии процесса становления личности и последовательные этапы этого процесса*.

Под сквозными линиями процесса становления личности при этом понимают такие его составляющие, которые, последовательно развиваясь, «пронизывают» все этапы образования, будучи обязательной составной частью каждой из них. Каждая из сквозных линий содержания образования разворачивается в дидактическую «цепочку», распределенную по годам и ступеням обучения. В частности, высшее профессиональное образование включает в себя теоретические предметы общепрофессионального и специального циклов, учебные практики и учебное проектирование. В теории содержания образования рассматривается также его градация по технологическому признаку: теоретическое обучение, практическое обучение, учебное проектирование [4].

Сквозная линия практического обучения подразделяется на последовательные ступени учебно-практической деятельности обучающихся, отобранные по содержанию и характеризующиеся адекватными формами и методами. Структура практического обучения детерминируется в части сквозных компонентов содержания структурой осваиваемой деятельности; в ее ступенчатом выражении – структурой поэтапного освоения деятельности.

К сквозной линии проектной подготовки студентов специальности «Информационные технологии в образовании» относятся учебные дисциплины:

- *Мировые информационные образовательные ресурсы*;
- *Психолого-педагогические основы проектирования информационных систем в образовании*;
- *Проектирование информационных систем в образовании*;

- *Мультимедиа технологии в образовании;*
- *Диагностика информационных образовательных ресурсов;*
- а также производственная практика.*

Покажем, что уже в первой из отмеченных дисциплин можно и нужно включать в учебный процесс элементы, нацеленные на формирование проектных знаний и умений.

К дидактическим единицам дисциплины **«Мировые информационные образовательные ресурсы»**, согласно ГОС ВПО специальности, относятся:

Мировые информационные образовательные ресурсы: определение, классификация и характеристика основных структур. Мировые информационные образовательные сети. Поисковые системы. Образовательные Web-сайты. Технология и практика взаимодействия пользователя с мировыми информационными образовательными ресурсами.

Дисциплина преподаётся в V семестре и имеет объём в 17 часов лекций и 17 часов практических занятий. Формой контроля успешности освоения дисциплины является зачет.

На практических занятиях, а также при выполнении домашних заданий студенты знакомятся с существующими информационными образовательными ресурсами, используемыми для их создания программными средствами. С точки зрения их подготовки к проектной деятельности принципиально важно, чтобы при этом знакомстве они учились оценивать их качество. Это пригодится им при изучении дисциплины «Диагностика информационных образовательных ресурсов», выполнении курсовых работ, учебном проектировании при прохождении производственной практики и подготовке выпускной квалификационной работы.

Следовательно, для обеспечения сквозной линии проектной подготовки будущих инженеров специальности «Информационные технологии в образовании» в учебную программу дисциплины «Мировые информационные образовательные ресурсы» целесообразно ввести раздел *«Критерии качества ЦИОР»*.

К числу этих критериев относятся [5, 6] следующие совокупности параметров:

- содержательные,
- технологические,
- дидактические,
- методические,
- дизайн-эргономические.

К числу *содержательных критериев* оценки качества цифрового информационного образовательного ресурса относятся следующие показатели:

- соответствие требованиям государственного образовательного стандарта по соответствующим направлениям подготовки специалистов;
- соответствие примерным программам учебных дисциплин федерального компонента;
- связь и отсутствие противоречий с основной учебной литературой, по соответствующим учебным дисциплинам.

К основным *технологическим критериям* относятся:

- надежность и устойчивая работоспособность;
- возможность работы на различных компьютерных средствах;
- надежность защиты от несанкционированных действий пользователей;
- эффективность использования ресурсов;
- простота в обновлении контента.

К числу *дидактических критериев* прежде всего те, которые предъявляются к учебным изданиям (учебникам, учебным и учебно-методическим пособиям). А именно:

- соответствие учебного материала современному уровню развития науки и техники;
- соответствия содержания, методов и форм обучения возрастным особенностям обучающихся, уровню их развития;
- преемственность знаний, полученных при изучении предшествующих дисциплин, междисциплинарные связи;
- точность определений и формулировок;
- соответствие поясняющих примеров и их достаточность;
- связь материала со специализацией основной образовательной программы;
- целесообразность и полнота практических рекомендаций;
- наличие и обоснованность причинно-следственных связей, нацеленных на активизацию самостоятельной работы студентов.

Кроме традиционных дидактических критериев для оценки качества ЦИОР применяются и специфические критерии, обусловленные использованием информационных и телекоммуникационных технологий для их создания и функционирования.

К числу дидактических критериев ЦИОР, обусловленных использованием современных информационных и телекоммуникационных технологий относят [5, 6] такие критерии как адаптивность, интерактивность, реализация возможностей компьютерной визуализации учебной информации.

С дидактическими критериями ЦИОР тесно связаны *методические критерии*. Последние характеризуют своеобразие и особенности предметной области ЦИОР, специфику соответствующей науки, ее понятийного аппарата, особенности методов исследования ее закономерностей; возможности реализации современных методов обработки информации и методологии реализации образовательной деятельности.

В ходе диагностики дидактических и методических свойств ЦИОР должна проводиться оценка полноты основных свойств ЦИОР, способствующих достижению педагогического эффекта, повышению результативности образования, оценка соответствия компонентов ЦИОР психологическим принципам и требованиям (возрастным особенностям и интересам обучающихся, использования развивающих компонентов обучения, способов активизации познавательной активности), оценка соответствия принципам вариативности образования.

При оценке *дизайн-эргономических критериев* пользовательский интерфейс ЦИОР анализируется с целью выявления:

- соответствие компонентов ЦИОР здоровьесберегающим требованиям;
- характеристики общей визуальной среды на экране монитора, цветовые характеристики, характеристики пространственного размещения информации;
- характеристики организации буквенно-цифровой символики и знаков на экране монитора;
- характеристики организации диалога (доступность для учащихся, время реакции на ответ или управляющее воздействие, число вариантов и правдоподобность ответов в вопросах типа “меню”, наличие инструкции или подсказки);
- характеристики звукового сопровождения (комфортность восприятия звуковой информации, удобство настройки звуковых характеристик, степень засоренности и оптимальность темпа звукового сопровождения);
- степень эстетичности компонентов ЦИОР.

Кроме того, при анализе ЦИОР необходимо давать оценку по следующим дизайн-эргономическим критериям:

- целесообразность, корректность и удобство использования клавиатуры и мыши;
- наличие и качество текста;
- наличие и качество статичных графических изображений;

- наличие и качество видеофрагментов и анимации;
- наличие корректирующей реакции на смысловые ошибки;
- удобство навигации;
- гармоничность цветовых и звуковых решений;
- наличие и эффективность работы поисковой системы;
- наличие контекстно-зависимой помощи;
- оптимальность расположения и размеры управляющих элементов;
- оптимальность выделения управляющих элементов цветом, формой, звуком;
- оптимальность визуальной среды (размеры объектов, расстояние между объектами, размеры шрифтов и др.)

Таким образом, для того, чтобы дисциплина *«Мировые информационные образовательные ресурсы»* была направлена на проектную подготовку, обучающиеся должны оценивать существующие МИОР, используя указанные критерии.

Поскольку проектирование – деятельность, то для обучения проектированию, безусловно, надо использовать деятельностные формы обучения, такие, как метод учебных проектов и любое проектирование как способ организации этой деятельности. Учебный проект позволяет организовывать целенаправленную деятельность обучающихся, в процессе которой актуализируются полученные на традиционных занятиях или каким-либо другим путем знания и навыки, приобретается личный опыт их практического применения, что способствует более глубокому усвоению знаний. Умение применить знания составляет неотъемлемую часть компетенции инженера. Поэтому помимо использования в проектной подготовке «сквозных линий» и особенностей объекта проектирования предлагается применять в этой подготовке методику деловых игр. Для решения задачи формирования необходимых компетентностей в рамках проектного обучения производится специальный подбор или разработка необходимых по содержанию и видам деятельности проектов.

В педагогике деловую игру рассматривают, прежде всего, как средство познания окружающей действительности. Деловые игры могут иметь разные цели, из которых главными являются изучение моделируемой системы и воспитание определённых качеств, умений и навыков, необходимых для успешной деятельности в системе.

В деловой игре сочетаются черты профессиональной и учебной деятельности. В ней моделируется содержание будущей профессии, требующее активного взаимодействия между членами группы. Эффективность деловых игр как средства познания обусловлена синтезом в одной деятельности игрового и познавательного интересов [7].

Рекомендуется [8] строить деловые игры на базе основных психолого-педагогических принципов:

- принцип имитационного моделирования конкретных условий и динамики производства;
- принцип имитационного моделирования содержания и форм профессиональной деятельности;
- принцип проблемности содержания имитационной модели и процесса её развёртывания в игровой деятельности;
- принцип ролевого взаимодействия в совместной деятельности;
- принцип диалогического общения;
- принцип двуплановости.

Главной особенностью деловой игры является то, что в результате совместной групповой деятельности обучающиеся не просто получают новые знания, а создают некоторый учебный продукт, материальный результат совместного труда, практически готовый к внедрению.

В процессе проведения деловой игры преподаватель подсказывает источники информации, ориентирует мысли обучающихся для самостоятельного поиска в нужном направлении. Обучающиеся должны самостоятельно решить проблему, получить реальный и осязаемый результат, действуя совместно, применяя имеющиеся у них знания из разных областей. Таким образом, вся работа над проблемой приобретает контуры проектной деятельности [6].

Существенно, что, так как проектный метод замкнут на конечный результат, то он обладает достаточно большой диагностической способностью.

Деятельность обучающихся при работе над проектом проходит те же этапы, что и при проведении научного исследования. К числу этих этапов относят [6]:

- определение проблемы и вытекающих из нее задач исследования,
- выдвижение гипотезы их решения,
- обсуждение методов решения и выбор одного из них,
- сбор данных,
- анализ полученных данных,
- решение проблемы
- оформление результатов,
- подведение итогов и формулировка выводов.

Деловая игра в области проектирования организуется по следующему типовому сценарию. На начальном этапе деловой игры происходит *формирование творческих групп* (3-5 человек). Каждой группе преподаватель даёт проектные задания. Обязанности внутри группы распределяются в зависимости от творческих интересов, интеллектуальных, организаторских способностей студентов. Руководит группой лидер, координирующий действия всех участников группы.

Проектная деятельность предполагает при работе над проектом соблюдения определенного алгоритма и сочетание различных видов деятельности: на разных этапах осуществления проекта решаются различные типы задач, выполняется соответствующий элемент проектной деятельности.

Выделим этапы проектирования ЦИОР любого уровня сложности:

- 1) этап стратегического планирования;
- 2) этап методического проектирования;
- 3) этап проектирования архитектуры;
- 4) этап реализации;
- 5) этап диагностики.

На *этапе стратегического планирования* происходит выработка общих целей и принципов создания цифрового информационного образовательного ресурса.

На *этапе методического проектирования* определяются методических принципов создания ЦИОР и его подсистем.

На *этапе проектирования архитектуры* согласуется структура и интерфейсы будущего ресурса, выбирается технического решения для его создания.

На *этапе реализации* исполняются согласованные стратегические, методические и технические решений в процессе создания ресурса.

На *этапе диагностики* тестируются элементы системы и их согласованность.

Проектирование ЦИОР в процессе проведения деловой игры предполагает выполнение следующих этапов: целеполагание, планирование, выполнение, защита, оценка.

Целеполагание - это подготовительный этап, на котором определяются цели проекта.

Планирование - это второй этап работы над проектом, на котором детально обсуждаются коллективные задачи группы и индивидуальные задачи каждого ее участника.

Выполнение - этап реализации проекта в соответствии с коллективными и индивидуальными задачами, поставленными перед участниками группы. Заключительной фазой этого этапа является подготовка к защите проекта, предполагающая создание сценария защиты и презентации проекта.

Защита проекта - этап, на котором все группы демонстрируют результаты своей работы. Основными *критериями успешности проекта* можно считать следующие:

- актуальность и значимость проекта,
- соответствие содержания заявленной теме,
- качественное представление материала,
- привлечение знаний из других областей,
- умение отвечать на вопросы аудитории,
- активность каждого участника группы.

Оценка деятельности групп - завершающий этап работы над проектом, проходящий в форме открытого обмена мнениями. Оценивание проходит с опорой на критерии успешности проекта.

Как отмечается в работе [9], одним из основных критериев научного исследования является продуктивность получаемого результата. Но при обучении проектированию большинства инженерных объектов (зданий, сооружений, самолётов, промышленных роботов и т.п.) оценка этого критерия затруднительна. Подготовка будущих инженеров специальности «Информационные технологии в образовании» является исключением из этого правила.

При обучении проектированию ЦИОР оценка критерия продуктивности осуществляется сравнительно просто. Получаемый результат можно продиагностировать. Если над одним заданием работает две и более микрогруппы, его можно сравнить с другими решениями, можно проранжировать альтернативные решения. И это могут сделать сами студенты.

Эта особенность ЦИОР позволяет заключить, что при подготовке будущих специалистов специальности «Информационные технологии в образовании» проектный метод обучения весьма эффективен.

Примерами тем заданий для проведения деловых игр в области проектирования ЦИОР являются:

- Электронное учебное пособие «Мировые информационные образовательные ресурсы»;
- Компьютерная обучающая программа по дисциплине «Тренинг общения»;
- Компьютерная тестирующая программа по дисциплине "Методика профессионального обучения";
- Информационно-справочная система «Работа в операционной среде Linux»;
- Интерактивная компьютерная программа по технологии верстки многополосной продукции.

Опыт работы, показывает, что применение проектного метода при подготовке инженеров специальности «Информационные технологии в образовании»:

- способствует формированию проектных знаний и умений,
- позволяет достаточно надёжно диагностировать сформированные знания и умения,
- выражается в получении кафедрой реальных программных продуктов.

Таким образом, в содержательной структуре подготовки будущих инженеров специальности «Информационные технологии в образовании» к проектной деятельности предлагается выделять *сквозную линию проектной подготовки*. Для её обеспечения целесообразно выделять критерии качества ЦИОР. Они пригодятся студентам при изучении некоторых дисциплин в дальнейшем, при выполнении курсовых работ, учебном

проектировании при прохождении производственной практики, при подготовке выпускной квалификационной работы.

При обучении проектированию большинства инженерных объектов оценка критерия продуктивности затруднительна. Подготовка будущих инженеров специальности «Информационные технологии в образовании» является исключением из этого правила. Эта особенность позволяет заключить проектный метод обучения весьма эффективен для обеспечения сквозной линии проектной подготовки. Существенно, что, так как проектный метод замкнут на конечный результат, то он обладает достаточно большой диагностической способностью.

Литература

1. Ширшов Е. В. Педагогические условия проектирования электронных учебно-методических комплексов: Монография / Е. В. Ширшов, О. В. Чурбанова. - Архангельск: Изд-во Архангельского государственного технического университета, 2005. – 307 с.
2. Компетентностный подход в образовании. /Под ред. Н. Ф. Радионовой. – СПб.: Издательство РГПУ им. А.И. Герцена, 2005. – 383 с.
3. Зеер Э. Ф. Ключевые квалификации и компетенции в личностно ориентированном профессиональном образовании // Образование и наука. 2005. №3(33) – С. 27 – 35.
4. Леднев В. С. Содержание образования: сущность, структура, перспективы. – М.: Высш. шк., 1991. – 224 с.
5. Краснова Г. А., Соловов А. В., Беляев М. И. Технологии создания электронных обучающих средств. - М.: МГИУ, 2001, - 223 с.
6. Моисеева М. В., Полат Е. С., Бухаркина М. Ю., Нежурина М. И. Интернет-обучение: технологии педагогического дизайна. М.: Издательский дом "Камерон", 2004. - 216 с.
7. Геронимус Ю. В. Игра. Модель. Экономика. – М.: Знание, 1989. – 208 с.
8. Вербицкий А. А. Активное обучение в высшей школе: контекстный подход. – М.: Высшая школа, 1991. – 204 с.
9. Леднев В. С. Научное образование: развитие способностей к научному творчеству. Изд. второе, исп. – М.: МГАУ, 2002. – 120 с.

СТРУКТУРА, НАПОЛНЕНИЕ И ПРОГРАММНАЯ РЕАЛИЗАЦИЯ ИНТЕРАКТИВНОГО САЙТА ГАЗЕТЫ ВУЗА

Д.А. Корнев, И.А. Лапшина, Т.С. Медведева

Рассмотрена структура сайта газеты на примере многотиражного издания «Университет ИТМО». Приводятся рекомендации по наполнению сайта. Описана программная реализация подобных проектов.

В настоящее время необходимым условием формирования общеевропейского пространства высшего образования является мобильность учащихся, преподавателей и сотрудников администрации образовательных учреждений. При этом большую роль играет мобильность не только в плане политики, экономики, общественной жизни, но и в культурном плане.

Применение открытых информационных систем, в том числе многотиражных вузовских газет, рассчитанных на использование всего массива информации, доступной в данный момент обществу в определенной его сфере, способствует не только ускорению научно-технического прогресса, интеллектуализации всех видов человеческой деятельности, но и созданию качественно новой информационной среды социума, обеспечивающей развитие творческого потенциала индивида.

В связи с широким внедрением в практику телекоммуникационных и компьютерных технологий многие образовательные учреждения стали информировать своих сотрудников о деятельности подразделений и коллективов не только через многотиражные периодические издания (газеты или журналы), но и через их электронные версии и сайты. Ориентируясь на цели образовательного учреждения, связанные, в том числе, и с формированием культуры личности обучающихся, и учитывая опыт работы редакции газеты университета, рассмотрим структуру, наполнение и программную реализацию интерактивного сайта газеты вуза.

В основе сайта газеты (см. рис. 1) лежит ее печатное издание, представленное электронной версией, например форматом PDF, передающей внешний вид печатного аналога.



Рис. 1. Структура сайта газеты

Далее, используя возможности электронного способа размещения информации, имеющийся материал дополняется:

- электронными версиями статей, оформленными с использованием гиперссылок,
- полными версиями статей, представленными в печатном аналоге в сокращенном варианте,

- аннотациями к статьям,
- интерактивным содержанием номера и т.д.

Сайт печатного издания имеет много дополнительных возможностей и путей расширения, направленных, главным образом, на организацию интерактивного общения с читателем и увеличение объема информационного материала. На сайте могут быть размещены следующие разделы:

- новости университета (редакции),
- история вузовской газеты,
- редакция газеты,
- электронные приложения, например, «Решения Ученого совета университета», «Карьера и трудоустройство», «Литературное приложение», «Петербургский стиль», «фотогалерея» и др.;
- электронная конференция и другие формы взаимообмена информацией, способствующие расширению читательской аудитории.

Цели сайта газеты соотносятся с целями общеобразовательного учреждения – формированием общей культуры личности обучающихся на основе внедрения информационных технологий:

- развитие мышления (наглядно-действенного, наглядно-образного, интуитивного, творческого, теоретического видов мышления);
- эстетическое воспитание (за счет использования возможностей компьютерной графики, технологии мультимедиа);
- развитие коммуникативных способностей;
- формирование умений принимать оптимальное решение или предлагать варианты решения в сложной ситуации (в ходе общения на электронной конференции);
- формирование информационной культуры, умений осуществлять обработку информации (при обсуждении статей номера и общения на электронной конференции);
- налаживание контакта преподавателей со студентами (путем общения на электронной конференции, в ходе обсуждения статей).

При разработке сайта газеты образовательного учреждения необходимо учитывать следующие факторы [1]:

- периодичность (частота выпуска газеты);
- объем представляемой информации;
- направленность сайта (информационный, развлекательный и т.д.);
- наличие архива газеты;
- степень интерактивности (учитывается частота обновления сайта, добавления новостей и т.д.);
- наличие электронной конференции: форумов, досок обсуждений и т.д. (в зависимости от целевой аудитории, популярности сайта и возможных путей его совершенствования).

При создании первого варианта сайта газеты «Университет ИТМО» основное внимание было уделено его структуре и дизайну.

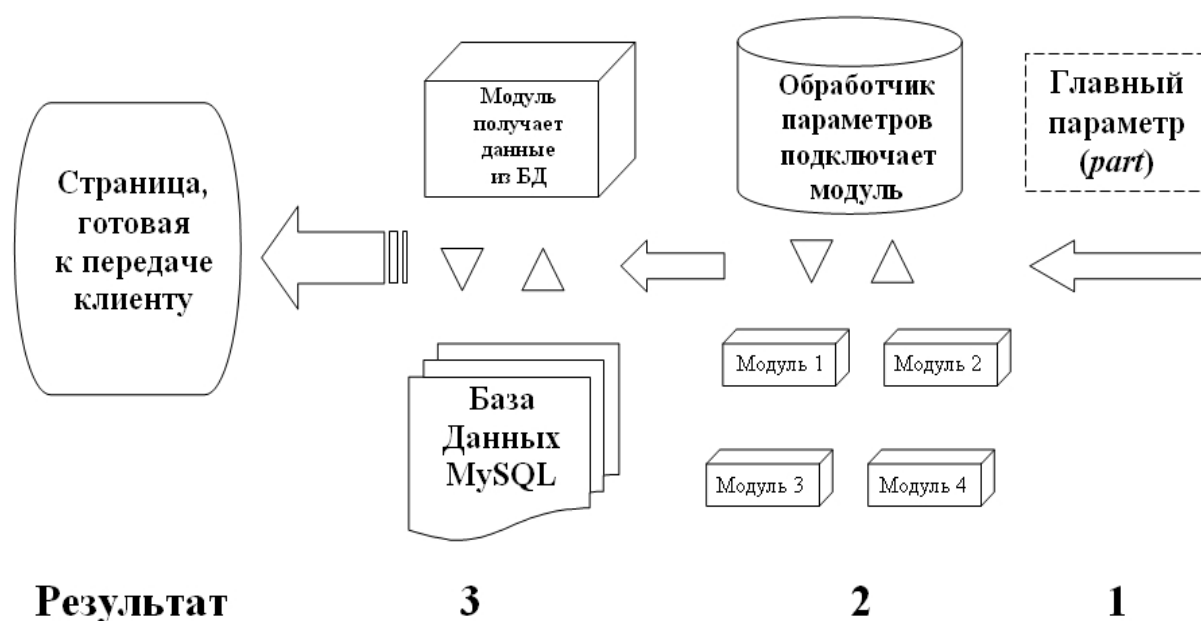
Структурно сайт делится на три информационных блока. Первый из них отвечает за обзорную информацию, необходимую пользователю для получения общего представления о сайте и его направленности. В него входят разделы: «Главная страница» и «Редакция». Второй блок представляет пользователю информацию о последнем номере и включает в себя разделы «Последний номер» и «Статья номера». Последний, третий блок дает читателю возможность общения как со сверстниками, так и с редакцией. На электронной конференции посетитель получает возможность завести интересующую тему или обсудить уже имеющуюся, также можно задать вопрос редакции или послать письмо.

Изначально сайт создавался при помощи языка HTML, без использования каких-либо баз данных. В процессе его внедрения и использования выяснилось, что данный подход неэффективен. Каждую статью последнего номера необходимо было прописывать вручную, вымерять и расставлять индивидуальные границы. К тому же каждый номер был доступен пользователям только до выхода нового, а затем удалялся. Раздел новостей также не был удобен в управлении.

Поэтому было решено, что для организации повседневной работы сайта необходимо использовать более совершенную динамическую технологию в комплексе системы управления базами данных. Этот механизм и был реализован с помощью языка WEB-программирования PHP и системы управления базами данных MySQL.

Со временем в газете увеличилось число публикаций, авторов, рубрик. Для привлечения к активному диалогу большего числа читателей и упорядочения значительного количества информационного материала было принято решение о расширении сайта за счет организации электронных приложений, соответствующих наиболее популярным рубрикам, новой структуры архива статей печатной версии, дополнительных элементов электронной конференции, способствующей интерактивному общению.

Вновь разработанный сайт редакции газеты с точки зрения программной реализации представляет собой модульную структуру (см. рис. 2), что позволяет в короткие сроки ввести в эксплуатацию новый модуль [2], например, приложение к сайту или рубрику.



Для ужесточения структуры сайта и ускорения интерпретации PHP-кода на сервере каждый модуль оформлен в отдельном файле. Благодаря этому из структуры исключены усложняющие ее условные ветки программы. Для каждого модуля в базе данных создано необходимое число таблиц, таким образом модуль представляет собой файл кода в комплексе с одной или несколькими таблицами.

Центральной частью структуры сайта является главный файл скриптов (index.php). В зависимости от переданных параметров формируются различные HTML-код, отправляемый пользователю. Главный параметр (англ. *part* – часть) отвечает за раздел, с которым происходит работа. Программа обращается к соответствующему модулю и передает другие параметры, тем самым завершая окончательное формирование HTML-кода.

Особенностью электронной версии газеты является возможность производить поиск статей по расширенным критериям – номеру газеты, автору, рубрике. Реализуется это с помощью созданной в базе данных сайта таблицы статей, которая содержит поля – внешние ключи на соответствующие таблицы со списками номеров, авторов и рубрик.

Используя возможности SQL-запроса, такие как LIKE, BETWEEN и др., можно задать только необходимые критерии поиска статьи [4]. Например, можно найти статью, опубликованную с номера 75 по 78, автором которой является Петр Иванов (выбирается из списка авторов, где каждому из них соответствует свой идентификатор), в названии которой содержится слово «техн», а в тексте «инновац». Кроме перечисленного, можно задавать даже позицию в содержании номера (например, найти все статьи второй полосы), или найти все «статьи номера» во всех номерах (признаком статьи номера является поле sign).

Важно отметить, что в созданном на портале университета разделе «Персоналии», в сведениях о некоторых сотрудниках приводятся ссылки на их публикации в газете (отдельные статьи). Описанная выше структура позволяет создать ссылку «просмотреть все публикации автора в многотиражной газете». Выбор статей в данном случае будет осуществляться по идентификатору автора.

Архив газеты отображает содержание таблицы номеров (*papers*). Эта страница содержит ссылки на оглавление каждого номера. Возможна сортировка архива по году выхода номера.

Газета вуза – это социально разностороннее издание, которое освещает различные области деятельности, в том числе такие, как трудоустройство, литературное и научное творчество, спортивную и общественную деятельность. С целью возможности увеличения объема публикуемого материала было принято решение о разработке на сайте приложений, например таких, как «Карьера и трудоустройство» и «Литературное приложение».

Приложение «Карьера и трудоустройство» включает в себя следующие возможности: просмотр опубликованных в газете статей по теме «Трудоустройство», общение на соответствующем разделе электронной конференции сайта и список материалов (страниц), представленных для внимания читателям сотрудником, ответственным за поддержку этого приложения.

Дополнительные материалы хранятся в базе данных, в таблице *pages*, модификация данных которой производится оператором через систему управления сайтом (СУС). Речь о ней пойдет позже.

Раздел «Литературное приложение» схож по структуре с разделом просмотра электронных версий статей газеты. Он включает в себя несколько таблиц. Основная их них – таблица текстов произведений (*texts_of_literary*) – содержит поля: автор произведения, жанр, отзывы (сортировка отзывов производится по идентификатору произведения), рецензии (других авторов) к произведениям (сортировка производится по идентификатору). Предусмотрена возможность предотвращения засорения таблицы отзывов с помощью графической библиотеки языка PHP GD2 [3].

Одной из важных задач сайта является организация интерактивного диалога с читателями. В ходе работы над проектом было принято решение использовать с этой целью свободно-распространяемую электронную конференцию *phpbb*. В конференции зарегистрированы все сотрудники редакции газеты. Они наделены правами модератора. Техническое обеспечение работы конференции осуществляет программист редакции. Контроль над содержанием форума осуществляет старший модератор – редактор сайта газеты.

Изменять содержание сайта возможно с любого компьютера, подключенного к сети Интернет. Модифицирование информации в базе данных производится с помо-

щью WEB-интерфейса – системы управления сайтом газеты. Система представляет связку PHP-скриптов, которые выполняют определенные операции с базой данных путем отправки SQL-запросов.

Пользовательский интерфейс системы управления реализован с помощью языка разметки гипертекста HTML. Оператор базы данных заполняет или изменяет различные HTML-формы, и после нажатия кнопки отправки формы браузер формирует HTTP POST запрос, который обрабатывается PHP-скриптом на стороне сервера [5].

Учитывая возможности, появившиеся в связи с развитием информационных технологий, оказалось возможным расширить традиционные рамки печатных изданий за счет создания на базе их электронных версий сайтов. В статье показано, что сайт газеты позволяет значительно расширить рамки печатного издания, увеличить читательскую аудиторию и привлечь к работе по его наполнению и программной реализации значительное число студентов образовательного учреждения.

На сайте возможна как публикация полных версий статей, сокращенных для печатного оригинала, так и дополнительных материалов в виде фотографий, стихов, комментариев. Расширение тематики публикаций возможно за счет создания на нем таких дополнительных информационных разделов, как приложения, например, «Литературное приложение», «Книгомания», «Карьера и трудоустройство» и др. В этих приложениях может не только помещаться дополнительная информация, но вестись ее обсуждение с выходом на форум.

Программная реализация интерактивного сайта постоянно совершенствуется с возрастанием объема публикуемого материала и позволяет, что очень важно, организовать в удобной для администратора форме активный диалог с читателями.

Литература

1. Лапшина И.А., Медведева Т.С. О совершенствовании сайта газеты «Университет ИТМО» на портале университета. // Научно-технический вестник СПбГУ ИТМО. Выпуск 32. Информационные технологии: теория, методы, приложения – СПб: СПбГУ ИТМО, 2006. – 322 с.
2. Мазуркевич А.М., Еловой Д.С. PHP: Настольная книга программиста. – Минск: Новое знание, 2003.
3. <<http://ru.php.net/>>
4. <<http://www.mysql.ru/docs/>>
5. <<http://www.mysql.org/>>

МЕТОДИКА ОБУЧЕНИЯ ПРОЕКТИРОВАНИЮ ЦИФРОВЫХ ОБРАЗОВАТЕЛЬНЫХ РЕСУРСОВ

Н.Н. Горлушкина, Н.О. Гордеева

Основной деятельностью выпускников СПб ГУ ИТМО специальности «Информационные технологии в образовании» должно стать проектирование цифровых образовательных ресурсов (ЦОР). В статье рассмотрены особенности проектирования цифровых образовательных ресурсов и методика обучения проектированию.

Введение

Под методикой обучения понимают совокупность средств, методов, принципов обучения, направленных на достижение определенных поставленных задач. Основной задачей специальности «Информационные технологии в образовании» должна стать подготовка и повышение квалификации инженерных кадров для отечественной системы образования. В образовательной программе специальности «ИТО» ключевую роль играют технические дисциплины, причем центральной является дисциплина «Проектирование информационных систем в образовании». Очевидно, именно она представляет основной инструмент будущего инженера-специалиста по разработке ЦОР.

Постановка задачи

В настоящее время разрабатывается методический комплекс в рамках дисциплины «Проектирование информационных систем в образовании», направленный на обобщение всех ранее полученных в рамках программы знаний студентов и совершенствующий их будущие профессиональные навыки. Для объективного формирования методического комплекса необходимо проанализировать опыт других специалистов.

Основные результаты исследований

Методика обучения проектированию ЦОР имеет ряд особенностей. Во-первых, умение проектировать должно тесно переплетаться с умением обучать, программировать, анализировать, графически оформлять. Во-вторых, проектирование ЦОР должно осуществляться с опорой на комплекс требований, составленный на основе критериев оценки эффективности ЦОР.

Методика обучения проектированию ЦОР должна подчиняться законам создания комплекса учебно-методического обеспечения дидактического процесса, при построении которого в наибольшей степени должны быть учтены различия в начальной подготовке обучаемых, варьироваться наглядность, полнота и конкретность подачи материала, обеспечиваться системность и вариативность представления информации, предусматриваться возможность проработки материала в свойственном каждому обучаемому темпе, упражнения в решении задач до получения запланированного результата. Это обеспечит адекватность ЦОР процессу овладения знаниями. [1]. Очевидно, что эти различия находятся в тесной зависимости от назначения проекта. В каждом конкретном случае будут предъявляться новые требования или желания заказчика – важно научиться подстраиваться под новую тематику. Все это возможно лишь при большом объеме практики.

В СПбГУ ИТМО дисциплина «Проектирование информационных систем в образовании» – новая, и поэтому методика находится на стадии разработки. Методический комплекс должен включать в себя:

- методическое пособие для студентов (в настоящее время разрабатывается выпускником специальности «Информационные технологии в образовании»);

- комплекс практических заданий;
- методические рекомендации по выполнению практических заданий;
- методические рекомендации для преподавателей дисциплины.

В настоящее время проводится изучение требований к ЦОР и отбор материалов для методического пособия. Анализируя опыт других специалистов, хочется выделить следующие рекомендации и мнения.

В [2] предлагается рассматривать проектирование обучающих программ как составной компонент общей исследовательской стратегии, которая предусматривает решение вопросов теории и технологии проектирования в комплексе с исследованием теории обучения и технологии компьютерного обучения. Проектирование обучающих программ рассматривается как многоуровневый процесс, и выделяются следующие уровни:

- концептуальный;
- технологический;
- операциональный;
- уровень реализации.

На концептуальном уровне задается модель обучения: обучение описывается как система, состоящая из двух подсистем – деятельности преподавателя и деятельности обучающихся. Все компоненты обучения, включая содержание (речь идет об общих принципах его проектирования) и метод обучения (описываемый на макроуровне, т.е. тип обучения), рассматривается в контексте этих деятельностей.

На технологическом уровне проект обучающей программы описывается в виде способа управления учебной деятельностью (метода обучения) на микроуровне. Здесь предписания переводятся на уровень технологии обучения.

На операциональном уровне процесс обучения описывается как решение дидактической задачи. На этом уровне указываются, во-первых, функции обучающей деятельности, возлагаемые на компьютер, и, во-вторых, основные способы (пути) управления учебной деятельностью.

Проект ЦОР на рассматриваемом уровне является основой для разработчиков системного программного обеспечения обучающей системы. Здесь задаются основные блоки программного обеспечения ЦОР через описание функций блоков – тех, которые непосредственно управляют учебной деятельностью (выдают обучающее воздействие), а также тех, которые обрабатывают данные о процессе решения учебной задачи, истории обучения, модели обучаемого и т.п.

Уровень реализации включает два уровня – педагогической и программной реализации [2].

Также при разработке ЦОР необходимо составить четкий алгоритм работы программы, включая различные варианты обратной связи и «откликов» компьютерной программы с учетом всевозможных вариантов действий пользователя.

Чтобы обеспечить эффективное использование ЦОР в учебном процессе, недостаточно заложить в программу систему указаний по работе с программой. Необходимо создавать также такие задания, которые учащийся захочет выполнять, а не поступать наоборот. Эффективное обучение сможет обеспечить только та обучающая программа, разработчики которой учитывают в достаточной степени человеческий фактор, видят в учащихся субъектов учебной деятельности, а не машину, слепо повинующуюся его указаниям [2].

В [3] выделяется два основных подхода к проектированию ЦОР – эмпирический и теоретический. Сторонники эмпирического подхода действуют методом проб и ошибок, делают упор на интуицию, здравый смысл, использование личного педагогического опыта, опыта разработки различных программных средств. Проектирование ЦОР

эмпирическим путём обычно идет от учебного предмета к обучающим воздействиям и завершается программной реализацией.

Проектирование ЦОР при теоретическом подходе осуществляется от проектирования образовательного процесса к технологии и методике обучения, и лишь затем осуществляется машинная реализация. Теоретический подход базируется на определенном фундаменте, включающем такие науки, как: педагогика, теория педагогического проектирования, педагогическая психология, инженерная и общая психология, информатика, кибернетика, теория высшей нервной деятельности, теория систем и др. [3].

При обучении студентов важно сразу обучить их проектировать, используя теоретический подход, чтобы создаваемые ими программы имели практический педагогический результат.

Одним из ключевых моментов в создании методики является определение четких критериев оценки качества ЦОР. Здесь важна определенность по двум причинам: во-первых, адекватная и качественная, аргументированная оценка курсовых работ студентов, а во-вторых, возможность выполнения курсовых и других работ студентами в соответствии с комплексом требований, составленным на основе критериев.

Критерии по оценке ЦОР составят основу методического пособия для студентов. Общая оценка цифрового ресурса будет складываться из баллов, проставленных по группам показателей анализа ЦОР:

- нормативный компонент,
- психолого-педагогический компонент,
- содержательный компонент,
- инструментальный компонент,
- процессуальный компонент.

Нормативный компонент включает в себя оценки по таким показателям, как соответствие нормативам и техническим стандартам, совместимость с аппаратно-программными комплексами различных конфигураций, наличие официальных заключений, соответствие содержания учебным программам, наличие свидетельств об официальной регистрации ЦОР, сертификатов (например, РОСИФОСЕРТ).

Психолого-педагогический компонент подразумевает получение баллов за наличие соответствия между способами представления информации в ЦОР и особенностями определенной возрастной группы (чтение текста, прослушивание текста, просмотр видеосюжета и др.), между формой предъявления информации в ЦОР и возрастными особенностями восприятия информации учащимися (например, размер шрифта, размер изображений, соответствующий объем «порции» учебного материала, цветовая гамма, возрастосообразный дизайн, возможность получить информацию с помощью различных органов чувств), между уровнем изложения учебного материала, включая тексты справочных материалов, динамическую помощь по работе с программным средством и возрастными категориями учащихся, за наличие доступности изложения и предъявления учебного материала с учетом возрастных особенностей группы учащихся.

В **содержательном компоненте** ЦОР должно быть оценено по уровню электронной поддержки (возможность использования при изучении определенной темы, раздела, полного курса одного учебного предмета; возможность использования в рамках совокупности учебных предметов), по научности изложения учебного материала с позиции системы образования (отсутствие фактических ошибок при изложении учебного материала, в текстах справочной помощи), по корректности изложения учебного материала с позиции системы образования (отсутствие грамматических и синтаксических ошибок в изложении учебного материала, в текстах справочной помощи; грамотная речь в звуковых фрагментах и т.п.), а также по наличию возможности организации индивидуальной образовательной деятельности учащегося (выполнение домашних заданий, творческих работ, самоконтроля и др.).

Оценка по показателям **инструментального компонента** складывается из наличия надежности работы ЦОР, соответствия современному технологическому уровню аналогичных ресурсов, возможности консультирования во время эксплуатации ЦОР, наличия и качества прилагаемых методических указаний и др.

Процессуальный компонент оценивает ЦОР по характеристикам удобства в использовании (легкость инсталляции, наличие вариативных модификаций, открытость для изменений, возможности в использовании).

По каждому из перечисленных показателей студенты должны стремиться получить максимальное количество баллов (в зависимости от показателя используется либо двухбалльная, либо пятибалльная шкала).

Заключение

Учитывая все указанные рекомендации, важно отметить, что требования к цифровым информационным ресурсам постоянно меняются, и для того, чтобы приблизиться к принятым нормам, необходимо начинать изучение дисциплины с обзора современных ЦОР и их анализа, так как многие показатели оцениваются именно в сравнении с существующими современными аналогами.

Методический комплекс по обучению проектированию ЦОР, составленный на основе критериев оценки качества ЦОР, позволит студентам создавать качественные информационные системы, а преподавателям – объективно оценивать их работу.

В настоящее время комплекс проходит частичную апробацию в рамках схожей дисциплины «Педагогические программные средства» у студентов специальности «Профессиональное обучение (охрана окружающей среды и природопользование)» с квалификацией «Педагог профессионального обучения» и постоянно дополняется новыми рекомендациями.

Литература

1. Образцов П.И. Психолого-педагогические аспекты разработки и применения в вузе информационных технологий обучения. / Монография. Орловский государственный технический университет. - Орел, 2000. - 145 с.
2. Елисеев М.Б. Обучающая программа по курсу «Педагогические программные средства» / Российский государственный профсоюзно-педагогический университет. – Екатеринбург, 1998.
3. Кречетников К.Г. Методология проектирования средств информационных технологий обучения / «ИТО-2001», ТОВМИ, Владивосток.

ВОЗМОЖНОСТЬ И ПЕРСПЕКТИВЫ ИСПОЛЬЗОВАНИЯ РЕСУРСОВ СИСТЕМЫ ДИСТАНЦИОННОГО ОБУЧЕНИЯ СПБГУ ИТМО ДЛЯ ФУНКЦИОНИРОВАНИЯ ВИРТУАЛЬНОГО ПРЕДСТАВИТЕЛЬСТВА СИСТЕМЫ ОТКРЫТОГО ОБРАЗОВАНИЯ

Е.А. Бердникова

В статье рассмотрен опыт использования ресурсов и сервисов системы дистанционного обучения СПбГУ ИТМО и возможность их интеграции с виртуальным представительством вуза в открытой информационной среде.

В настоящее время на базе технологий дистанционного обучения формируются специфические информационные образовательные среды отдельных вузов. В перспективе совокупность таких сред за счет ряда общесистемных соглашений сможет обеспечить доступ к образовательным ресурсам на региональном и федеральном уровнях, создавая тем самым основу системы открытого образования (ОО).

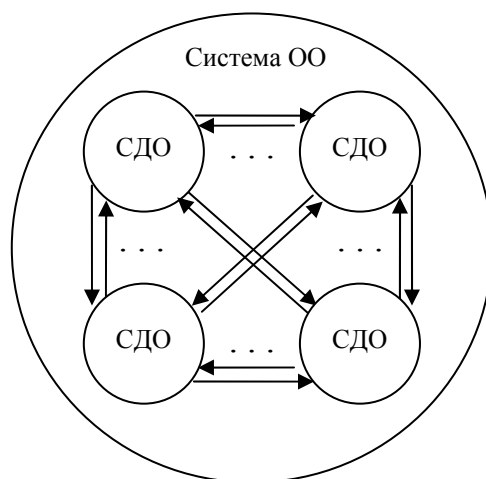


Рис.1. Система ОО

При создании системы открытого образования целесообразно использовать накопленные в высшей школе научно-методические материалы, информационные ресурсы и технологии, опыт проведения дистанционного обучения, существующую телекоммуникационную инфраструктуру и организационные структуры высшей школы.

В общем случае интегрированная система открытого образования, представленная на рис. 1, призвана реализовать следующие принципы [1]:

- *децентрализация* – предоставление образовательным учреждениям федерального и регионального уровня права самостоятельно решать вопросы организации учебного процесса в системе ОО и создание условий взаимовыгодного обмена образовательными продуктами и услугами;
- *демократизация* – создание равных прав всем образовательным учреждениям в решении нормативно-правовых вопросов и вопросов коммерциализации учебного процесса в системе ОО;
- *глобализация* – открытость информационных ресурсов и организации учебных процессов во всех точках системы ОО на веб-серверах для глобального взаимодействия и обмена передовым опытом;

- *регионализация* – региональные соглашения, объединения и группировки для содействия торговой и экономической интеграции в качестве средств усиления их конкурентоспособности.
- *непрерывность* - обеспечение в ОО всех уровней образования, принятых в системе непрерывного образования Российской Федерации: начального, среднего, среднего профессионального, довузовской подготовки, высшего, дополнительного, послевузовского;
- *интеграция* – создание виртуальной электронной библиотеки учебных курсов, банков данных и баз знаний, распределенных по отраслям знания, с защитой авторских прав.

Принципы построения и методы реализации системы ДО СПбГУ ИТМО делают ее открытой и универсальной [2]. Это обеспечивается тем, что она предоставляет преподавателям неограниченные возможности для разработки электронных учебно-методических комплексов (УМК) по любым дисциплинам с использованием открытых форматов. Благодаря этим свойствам система ДО СПбГУ ИТМО соответствует всем основным принципам, на которых базируется система открытого образования. Тем самым университет посредством ДО сможет стать полноценным участником общенациональной интеграции информационных образовательных сред.

Структура электронного УМК системы ДО основывается на учебных планах и рабочих программах дисциплин, разработанных в соответствии с государственными образовательными стандартами. Электронные учебные курсы состоят из преподавателями из набора элементов, в который входят конспекты, тесты, виртуальные лаборатории, практикумы и информационные ресурсы. Сервисы и инструментарий системы ДО, показанные на рис. 2, обеспечивают возможность общения (в том числе в режиме real-time) между преподавателями и студентами, просмотра успеваемости и коррекции учебных материалов.

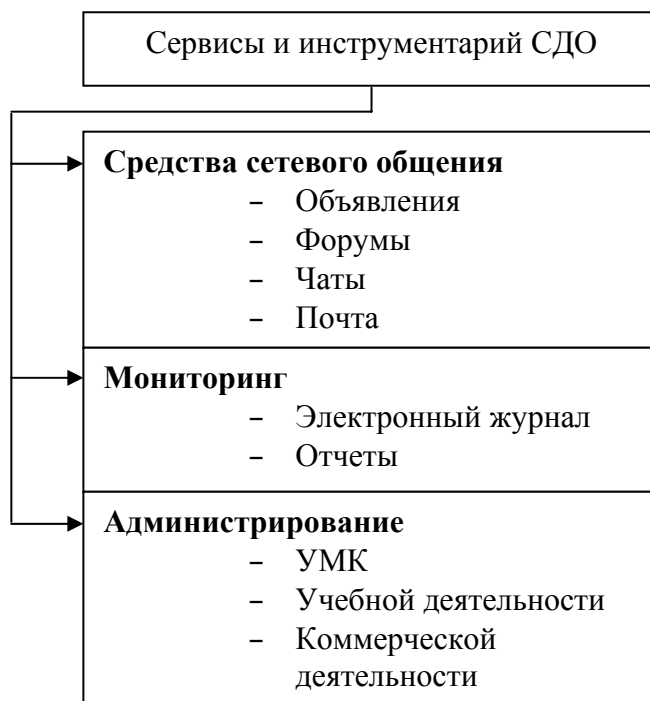


Рис..2. Сервисы и инструментарий системы дистанционного обучения СПбГУ ИТМО

На данный момент в системе ДО СПбГУ ИТМО размещено более 114 электронных курсов по всевозможным дисциплинам. Эти курсы содержат различные типы элементов, реализованных в системе ДО (рис. 3).

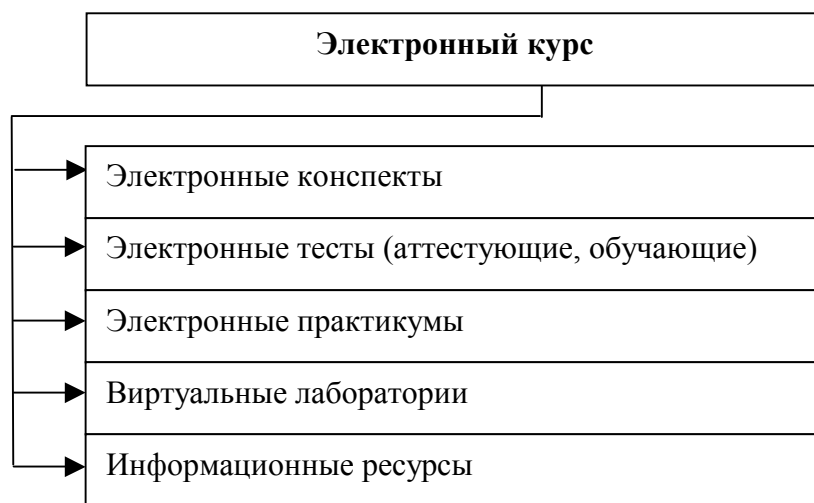


Рис. 3. Структура электронного курса

В системе реализована возможность взаимосвязи элементов различных типов при составлении электронного курса. Например, студент не получает доступ к электронному практикуму, не выполнив аттестующий тест. При чтении электронного конспекта учащийся может выполнить виртуальную лабораторию по изучаемому разделу. Подобные траектории изучения материала составляются авторами электронных курсов, что обеспечивает вариативность в изучении дисциплины.

Из рассмотренного видно, что, помимо серьезных перспектив использования технологий системы в становлении единой информационной образовательной среды, открываются широкие возможности по обеспечению виртуального представительства СПбГУ ИТМО в системе открытого образования учебно-методическими ресурсами и сервисами системы дистанционного обучения.

Литература

1. Создание системы открытого образования / Приказ № 2925 Министерства образования Российской Федерации. <<http://www.informika.ru>>.
2. Техническое руководство по разработке учебно-методического комплекса для системы дистанционного обучения (версия 1.54)/ – СПб.: СПбГУ ИТМО, 2006. – <<http://cde.ifmo.ru/index.php?node=10>>.

ПРОБЛЕМЫ ПОДГОТОВКИ ПРОФЕССИОНАЛЬНЫХ КОНСУЛЬТАНТОВ В ОБЛАСТИ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И ПУТИ ИХ РЕШЕНИЯ

С.В. Кессель

Разработана методика подготовки профессиональных консультантов в области информационных технологий в университетах технического типа с учетом их особенностей. Показано, что в настоящее время подготовка профессиональных консультантов в области информационных технологий не ведется ни в одном вузе. Обоснована актуальность исследования, сформулированы цель и гипотеза исследования. Апробация результатов проводилась в рамках курса «Профессиональное консультирование» для студентов IV курса естественнонаучного факультета (ЕНФ) СПбГУ ИТМО.

В настоящее время в связи с бурным развитием информационных технологий заметно повысился спрос на высококлассных специалистов, осуществляющих внедрение программных продуктов. Так как деятельность таких специалистов связана с консультированием (консалтингом), то их принято называть профессиональными консультантами. В данном случае речь идет о профессиональных консультантах в области информационных технологий (ИТ-консультантах). Так как значительная часть профессиональной деятельности консультанта связана с передачей знаний (в частности, с обучением пользователей), то он должен не только обладать широкими знаниями в предметной области (информационные технологии и, в частности, бизнес-системы), но и иметь подготовку в области психологии и педагогики.

В настоящее время в классификаторе квалификаций квалификации «профессиональный консультант» нет, подготовка профессиональных консультантов в российских вузах не ведется. Некоторый опыт подготовки профессиональных консультантов в области информационных технологий накоплен на кафедре технологий профессионального обучения СПбГУ ИТМО. Здесь при участии автора поставлена дисциплина «Профессиональное консультирование». Указанная дисциплина преподается в цикле дисциплин специализации студентам двух специальностей: 02.02.00 – профессиональное обучение (информатика, вычислительная техника и компьютерные технологии); 03.05.00 – информационные технологии в образовании. Статистика трудоустройства выпускников кафедры за последние 5 лет показывает, что из 95 выпускников 57 работают профессиональными консультантами (или выполняют функции профессиональных консультантов) в области информационных технологий.

К сожалению, если по профессиональному консультированию в области управления (управленческому консультированию) имеется достаточно большое число руководств [1–6], том числе иностранных авторов [7, 8], то по профессиональному консультированию в области информационных технологий аналогичные руководства практически отсутствуют. Это свидетельствует о *необходимости и своевременности* разработки методики подготовки профессиональных консультантов в области информационных технологий, т.е. об *актуальности* настоящего исследования.

Сказанное позволяет сформулировать *проблему*, состоящую в разработке методики подготовки профессиональных консультантов, базирующейся на системном подходе. Решение этой проблемы связано с противоречиями:

- между необходимостью подготовки профессиональных консультантов (в частности, в области информационных технологий) и отсутствием соответствующих специальностей высшего профессионального образования;
- между возможностью подготовки профессиональных консультантов в рамках специализаций некоторых специальностей высшего профессионального образования и недостаточной разработкой методики этой подготовки;

- между необходимостью подготовки профессиональных консультантов, обеспечивающих внедрение и сопровождение информационных технологий, и полным отсутствием методики этой подготовки, базирующейся на системном подходе.

Актуальность, проблема и отмеченные противоречия послужили основанием для формулировки *темы исследования* «Методика подготовки профессиональных консультантов в области информационных технологий».

Объектом исследования выступает процесс подготовки профессиональных консультантов в области информационных технологий.

Предмет исследования – методика подготовки профессиональных консультантов в области информационных технологий.

Цель исследования – разработка методики подготовки профессиональных консультантов в области информационных технологий, базирующейся на системном подходе.

Гипотеза исследования заключалась в том, что успешность подготовки профессиональных консультантов в области информационных технологий будет обеспечена, если:

- она будет проводиться со студентами специальности «Информационные технологии в образовании», при условии специализации основной образовательной программы «Профессиональное консультирование»;
- в качестве дисциплин специализации будущие инженеры освоят дисциплину «Технологии профессионального консультирования»;
- в качестве учебных практик студенты пройдут практический тренинг по профессиональному консультированию в рамках обучения на рабочем месте;
- курсовое и дипломное проектирование студентов, оставаясь в общих чертах согласованным с требованиями Государственного образовательного стандарта специальности «Информационные технологии в образовании», будет связано с решениями тех или иных задач профессионального консультирования в области информационных технологий;
- в процессе диагностики успешности освоения дисциплин специальности проверяются не только знания и умения, регламентируемые Государственным образовательным стандартом по соответствующим дисциплинам, но и знания и умения, необходимые для работы профессиональным консультантом;
- будущие инженеры в дополнение к основной образовательной программы специальности «Информационные технологии в образовании» осваивают дополнительную образовательную программу получения дополнительной квалификации «Преподаватель высшей школы».

Исходя из поставленных целей, в ходе исследования решались следующие *задачи*.

Первая группа задач – определение исходных принципов построения программы дисциплины:

- рассмотрение профессионального консультирования в области информационных технологий: содержание, технологии, необходимый уровень;
- анализ государственного образовательного стандарта специальности «Информационные технологии в образовании»;
- подготовка содержания дисциплин, необходимых для подготовки профессиональных консультантов.

Вторая группа задач – построение методики подготовки профессиональных консультантов в области информационных технологий:

- определение содержания базовых дисциплин;
- подготовка методики преподавания базовых дисциплин;
- разработка содержания учебных практик, курсового и дипломного проектов;

- разработка методики диагностики знаний и умений.

Третья группа задач:

- проведение констатирующего и формирующего экспериментов;
- обработка и анализ результатов эксперимента.

Для решения задач исследования использовались следующие *методы исследования*:

- анализ философской, научно-методической, психолого-педагогической литературы по проблемам профессионального консультирования, в частности в области информационных технологий;
- анализ научной литературы по педагогике, информатике, бизнесу;
- анализ государственных образовательных стандартов по ряду специальностей, зарубежных и отечественных программ подготовки профессиональных консультантов или специалистов, близких к ним;
- метод экспертных оценок, методы статистической обработки результатов эксперимента.

Научная новизна исследования заключается в разработке методики подготовки по новой специализации, включающей в себя как специальные знания, так и педагогические и практические навыки.

Теоретическая значимость исследования состоит в том, что обоснованы теоретические положения выбора методики обучения будущих профессиональных консультантов:

- уточнено понятие «профессиональное консультирование»;
- на основе сформулированных определений термина «профессиональное консультирование» сформулирован ряд новых принципов отбора содержания обучения будущих профессиональных консультантов;
- построена методика подготовки будущих профессиональных консультантов.

Практическая значимость исследования заключается в том, что на основе построенной методической теории подготовки консультантов могут быть созданы варианты учебной дисциплины «Профессиональное консультирование» для университетов технического типа с учетом их особенностей.

Достоверность и обоснованность полученных результатов обеспечена: методологией исследования, теоретическим обоснованием положений исследования и практической реализацией отдельных элементов построенной методической теории обучения; количественным и качественным анализом результатов исследования, полученным на основе использования методов исследования, адекватных предметным задачам и этапам исследования; использованием методов математической статистики для обработки результатов проведенного опытно-экспериментального исследования.

Апробация результатов исследования осуществлялась через публикации и выступления на научных конференциях, а также при чтении курса «Профессиональное консультирование» студентам IV курса ЕНФ СПбГУ ИТМО.

Внедрение результатов исследования проводилось в курсе «Профессиональное консультирование» для студентов IV курса ЕНФ СПбГУ ИТМО.

Экспериментальное исследование согласно поставленным задачам проводилось в период с 2005-2006 гг. в курсе «Профессиональное консультирование» для студентов IV курса специальности «Профессиональное обучение» СПбГУ ИТМО (2004/05–2005/06 уч. гг.);

На первом и втором этапах педагогического эксперимента (*констатирующий и поисковый эксперимент*) проводились сбор и анализ необходимой для уточнения гипотезы исследования информации. На третьем этапе (*формирующий эксперимент*) строилась теоретическая модель методической системы подготовки будущих профессиональных консультантов и осуществлялась эмпирическая проверка ее эффективности.

В ходе *констатирующего эксперимента* изучалось состояние проблемы подготовки профессиональных консультантов, состояние преподавания вопросов, связанных

с профессиональной деятельностью таких специалистов в технических вузах; необходимость построения методической системы подготовки профессиональных консультантов, обеспечивающей будущего специалиста знаниями в необходимых предметных областях. В связи с этим были исследованы Государственный образовательный стандарт высшего профессионального образования по специальности «Информатика», а также учебные пособия для университетов как технического, так и экономического типа.

Третий этап педагогического эксперимента (*формирующий эксперимент*) был посвящен проверке гипотезы исследования. Цель третьего этапа экспериментального исследования (формирующий эксперимент) состояла в проверке выдвинутой гипотезы. На основе предложенной теоретической модели методической системы обучения нами был построен один из возможных вариантов учебного курса «Профессиональное консультирование» для студентов IV курса ЕНФ СПбГУ ИТМО. В соответствии с выдвинутой гипотезой проверка целесообразности теоретических положений разработанной методики обучения студентов заключалась в выяснении, влияет ли выбранный подход к отбору содержания обучения на совершенствование подготовки будущих профессиональных консультантов. На основе выбранной цели обучения задачу формирующего эксперимента можно сформулировать следующим образом: определить, повысился ли уровень знаний будущих профессиональных консультантов и будет ли способствовать применению предложенной методической системы обучения приобретению ими фундаментальных знаний в предметной области. Проведенный эксперимент подтвердил верность выдвинутых теоретических положений.

Таким образом, в рамках проведенного исследования было выявлено отсутствие в классификаторе квалификаций квалификации «Профессиональный консультант», показано, что подготовка профессиональных консультантов в российских вузах в настоящее время не ведется. Указанное несоответствие показало актуальность проводимого исследования. В рамках работы были выявлены объект, предмет исследования, сформулирована цель исследования и гипотеза исследования. В результате оказалось возможным разработать методику подготовки профессиональных консультантов в области информационных технологий и опробовать ее на практике.

Литература

1. Блинов А.О., Бутырин Г.Н., Добренкова Е.В. Управленческий консалтинг корпоративных организаций / Учебник. – М.: Инфра-М, 2002..– 192 с.
2. Гульковский А.В., Рысюк В.Н.. Управленческое консультирование. Вопросы и ответы. – М.: Юркнига, 2004. – 288 с.
3. Деева Е.М., Васильев Г.А. Управленческое консультирование. / Учебное пособие для студентов вузов, обучающихся по специальности 061100 (Менеджмент организации). – М.: ЮНИТИ-ДАНА, 2004. 255 с.
4. Капустин П. Управленческое консультирование для руководителей. Практическое пособие. – СПб: Бизнес-пресса, 2000. 160 с.
5. Конов Д., Куликов А. Управленческое консультирование. // Консультант директора. 1997. №3. – С. 8–11.
6. Консалтинговые услуги в условиях реформирования экономики России / Центр исслед. и содействия развитию лесного комплекса. / Ред. М. И. Кныш. – СПб: Дмитрий Буланин, 2003..–191 с.
7. Макхэм К. Управленческий консалтинг / Пер. с англ. – М: Дело и Сервис, 1999. – 288 с.
8. Управленческое консультирование. - Под ред. М. Кубра. - М, "Интерэксперт", 1992 - 319 с.

ЭНЕРГОИНФОРМАЦИОННЫЕ ПОТОКИ ПРИ ОБУЧЕНИИ И ИХ КОЛИЧЕСТВЕННЫЕ ХАРАКТЕРИСТИКИ

И.Б. Стражмейстер

Процесс обучения рассматривается как энергоинформационный обмен между обучающим и обучающимися субъектами. Носителями и приемниками энергоинформационных потоков являются люди, которые представляются как термодинамическая система, далекая от состояния равновесия и обладающая открытостью, нелинейностью, когерентностью. В качестве количественных характеристик потоков при обучении принимаются описанные в статье термодинамические показатели.

Введение

Процесс обучения характеризуется созданием энергоинформационного потока между преподавателем и студентами, посредством которого происходит передача знаний. Для оценки происходящих при этом процессов предлагается применить способ определения локального удельного потока энтропии человека [1]. С тех пор, как возникло научное знание, начали формироваться науки, разрабатываться научные методы, одним из объектов исследования, естественно, был сам человек. Особенно перспективными являются приборные методы, использование которых позволило получить новые знания не только о физиологических и анатомических процессах в организме человека, но и о его психических состояниях. В последнее время развиваются такие науки, как синергетика и энергоинформатика [2]. Первая рассматривает процессы самоорганизации, перехода в открытых нелинейных средах различной природы от хаоса к порядку и обратно; вторая – энергоинформационные взаимодействия в природе.

Возможность расширения числа приборных методов исследования человека представляет в настоящее время термодинамика открытых систем, далеких от состояния равновесия. Термодинамические методы исследования постоянно расширяют сферу своего применения. В связи с развитием синергетики в конце XX века энтропия широко используется для исследования открытых термодинамических систем, далеких от состояния равновесия и обладающих, в частности, такими свойствами, как нелинейность и когерентность. Особый интерес представляет функция диссипации при учете сопряженных процессов при обмене энергией в открытых, нелинейных, когерентных системах, которыми являются и биологические объекты, в частности, человек как субъект процесса обучения.

Постановка задачи

Одним из перспективных направлений термодинамических исследований является изучение энергоинформационных потоков человека под воздействием различных внутренних и внешних факторов, в частности, в процессе обучения. Исследования энергоинформационных воздействий на человека проводятся в Центре энергоинформационных технологий при кафедре компьютерной теплофизики и энергофизического мониторинга СПбГУ ИТМО под руководством д.т.н., профессора Г.Н. Дульнева [3, 4]. Для оценки реакции человека на информационное воздействие процесса обучения предлагается такой показатель, как энтропия, которая является мерой характеристики перехода системы от хаоса к порядку и обратно в процессе обмена с окружающей средой. Поскольку исследуется изменение энергоинформационных потоков в процессе обучения и полученные результаты сравниваются с показаниями фона и последствием, рассчитывается скорость изменения энтропии, или функция диссипации. Изменение энергоинформационных потоков человека в процессе обучения выводится в виде зависимости коэффициента хаоса от времени до информационного воздействия, в процессе

обучения и после него. Сравниваются показатели группы лиц, участвующих в процессе обучения.

Количественное определение энергоинформационных потоков при обучении

Понятие энтропии впервые было введено в термодинамике для определения меры необратимого рассеяния энергии. Энтропия – параметр, определяющий состояние системы с точки зрения ее внутренней упорядоченности и являющийся функцией этого состояния: чем больше хаос в системе, тем выше ее энтропия [5].

По определению Клаузиуса, изменение энтропии ΔS равно сообщенной теплоте ΔQ , отнесенной к абсолютной температуре T .

$$\Delta S = \frac{\Delta Q}{T}; \quad dS = \frac{\partial Q}{T} \quad (1)$$

Энтропия является функцией состояния, как и давление, температура, объем, внутренняя энергия, т.е. любому состоянию можно сопоставить вполне определенное значение энтропии. При абсолютном нуле температуры по теореме Нернста энтропия тоже равна нулю.

Энтропия открытой системы постоянно изменяется под влиянием внешних условий, воздействующих на систему: внешние условия определяют микросостояния, доступные системе, и их число. В пределах доступных для нее микросостояний система достигает равновесия, а энтропия, следуя за изменениями внешних условий, достигает своего максимального значения. Чем более сильно упорядочена система, тем меньше число микросостояний, которыми осуществляется макросостояние.

Австрийский физик Л. Больцман, изучая статистическое распределение молекул в определенном объеме газа, вывел формулу, по которой энтропия S определяется логарифмом числа микросостояний P , посредством которых реализуется рассматриваемое макросостояние (статистическим весом системы) [6]

$$S = k \ln P, \quad (2)$$

где $k = R/N = 1,38 \cdot 10^{-23}$ Дж/К – фундаментальная мировая постоянная Больцмана; $R = 8,31$ Дж/(моль·К) – молярная газовая постоянная; $N = 6,06 \cdot 10^{23}$ моль⁻¹ – число Авогадро. Менее упорядоченное состояние (большой хаос) имеет больший статистический вес, так как оно может быть реализовано большим числом способов. Следовательно, энтропия – мера неупорядоченности системы. Порядок создается искусственно, беспорядок – самопроизвольно, так как ему отвечает большая вероятность, большая энтропия. Разумная деятельность человека, в частности, процесс обучения, направлена на преодоление неупорядоченности.

В открытых системах, в частности, в биологических, обменивающихся с окружающей средой массой, энергией и информацией, однородное состояние равновесия может терять устойчивость и необратимо переходить в неоднородное стационарное состояние, устойчивое относительно малых возмущений. В таких системах обнаружен эффект движения от хаоса к порядку, эффект самоорганизации. Вопрос состоит в том, как связано изменение энтропии с параметрами процессов в открытой системе и можно ли предсказать общее направление необратимых процессов в открытой системе по изменению ее энтропии. Указанная связь была установлена И. Пригожиным [7] и выражается в том, что общее изменение энтропии dS открытой системы может происходить независимо либо за счет процессов обмена с внешней средой ($d_e S$), либо вследствие внутренних необратимых процессов ($d_i S$):

$$dS = d_e S + d_i S. \quad (3)$$

В классической термодинамике рассматриваются главным образом равновесные состояния системы, в которых параметры не изменяются во времени. Однако в открытых системах, в частности, биологических, реакции и соответствующие энергетические

превращения происходят постоянно, и поэтому здесь необходимо знать скорости трансформации энергии в каждый момент времени. Это значит, что в энергетических расчетах надо учитывать как фактор времени, так и фактор пространства. Для этого необходимо каким-то образом сочетать термодинамический и кинетический подходы в описании свойств открытой системы. При этом необходимо рассматривать скорость изменения энтропии – функцию диссипации σ , которая описывает производство энтропии $d_i S$ в единицу времени τ в единице объема V :

$$\frac{d_i S}{d\tau} = \int_v \sigma dV. \quad (4)$$

Поскольку формула (4) учитывает время, то введение функции диссипации дает возможность перехода от равновесных процессов, протекающих очень медленно (где на каждом этапе достигается состояние равновесия) к неравновесным сопряженным процессам. По первому закону Бауэра все живые системы стремятся к устойчивому неравновесию, а косные – к равновесию.

В соответствии с соотношением Онзагера между движущими силами и скоростями (потоками) существует взаимосвязь, при которой увеличение (уменьшение) движущей силы вызывает соответствующее увеличение (уменьшение) скорости процесса. Это относится ко всем необратимым сопряженным процессам, которыми и являются перенос массы, энергии и информации. Во всех этих случаях возрастание энтропии (производная энтропии по времени, или функция диссипации) имеет вид

$$\sigma = J_1 X_1 + J_2 X_2 + \dots + J_n X_n = \sum_{i=1}^n J_i X_i, \quad (5)$$

где X – движущая сила, причина, вызывающая процесс; J – величина потока, вызванного данным процессом, следствие процесса. Таким образом, функция диссипации включает в себя различные сопряженные процессы обмена открытой термодинамической системы с окружающей средой, такие как обмен массой, энергией и информацией.

Основные результаты исследований

В такой открытой и неравновесной системе, как обучающий–обучающиеся, происходит в основном информационный обмен [8]. Количественная оценка переноса информации от обучающего к обучающимся осуществляется с помощью регистрации плотности локального потока и температуры определенных участков кожи участников исследования с последующим пересчетом их в коэффициент хаоса [9]. Результаты исследований представляются в виде графиков изменения коэффициента хаоса до, в процессе и после обучения для группы лиц, участвующих в учебном процессе.

Измерения реализуются преобразователями теплового потока (тепломеры Геращенко) со встроенными медно-константановыми термопарами с помощью лабораторного измерительного комплекса «Эниотрон-3», позволяющего проводить комплексные измерения полей различной физической природы. Тепломер Геращенко обладает двойным электрическим слоем, который позволяет регистрировать как энергетические (тепловые), так и информационные воздействия [10, 11]. Тепломер помещается в так называемых биологически активных точках тела испытуемого [1], с которых производится съем информации.

Значения теплового потока $q(\tau)$ с площади A температуры T для промежутка времени $\Delta\tau$ позволяют получить в реальном масштабе времени текущие значения удельной термодинамической энтропии ΔS на выбранном участке тела по известной формуле

$$\frac{dS}{d\tau} = \frac{q(\phi) \cdot \text{Дж}}{T}, \text{ К} \cdot \text{м}^2, \quad (6)$$

где $q(\tau)$ – плотность потока с площади поверхности тела A за промежуток времени $\Delta\tau$, T – абсолютная температура датчика. Для открытой системы, которой является человек, полная энтропия вычисляется по формуле (3): складывается из производства энтропии d_iS и обмена энтропией с окружающей средой d_eS . Величина производства удельной энтропии для человека (средняя площадь поверхности тела которого около 2 м^2) хорошо изучена и составляет $d_iS = 0,16 \text{ Дж}/(\text{м}^2 \cdot \text{К})$ [6]. Поэтому величину обмена удельной энтропии находят по формуле:

$$d_eS = dS - d_iS = dS - 0,16. \quad (7)$$

Для определения упорядоченности состояния системы используются меры порядка Π и хаоса X . Согласно Л. Бриллюэну [10], степень порядка в системе равна разнице между максимальным $d_eS_{\max}$ и текущим d_eS значениями энтропии, а степень хаоса – разнице между текущим d_eS и минимальным $d_eS_{\min}$ значениями энтропии:

$$\Pi = d_eS_{\max} - d_eS, \quad (8)$$

По аналогии с формулой (8) выводим выражение для величины хаоса X системы:

$$X = d_eS - d_eS_{\min}. \quad (9)$$

Действительно, для случая $d_eS = d_eS_{\min}$ степень хаоса равна нулю, для случая $d_eS = d_eS_{\max}$ степень порядка равна нулю, что и следовало ожидать.

Для количественной оценки информационных процессов, таких как обучение, предлагается использовать относительные нормированные оценки хаоса и порядка. Для этого значения хаоса X и порядка Π делят на общий знаменатель, представляющий собой разницу между максимальным $d_eS_{\max}$ и минимальным $d_eS_{\min}$ значениями приращения обмена энтропией с внешней средой, и для оценки состояния сознания используются сами коэффициенты K_x или K_n .

Оценки хаоса или порядка предлагается определять по следующим формулам:

$$K_x = \frac{d_eS - d_eS_{\min}}{d_eS_{\max} - d_eS_{\min}}, \quad K_n = \frac{d_eS_{\max} - d_eS}{d_eS_{\max} - d_eS_{\min}}, \quad K_x + K_n = 1, \quad (10)$$

где d_eS , $d_eS_{\max}$, $d_eS_{\min}$ – текущее, максимальное и минимальное приращение удельной термодинамической энтропии обмена с внешней средой. Можно считать, что масса и энергия изменяются по линейному закону, и в производной энтропии, вызванные данными процессами (функции диссипации), являются постоянными величинами и сокращаются при расчетах коэффициентов хаоса или порядка. Таким образом, можно предположить, что в графическом виде зависимость коэффициента хаоса от времени отражает именно динамику информационных процессов, возникающих во время обучения.

Апробация результатов производилась на базе СПбГУ ИТМО. На кафедре КТФ и ЭМ проводились исследования изменения энергоинформационных потоков в процессе устного счета, анализа сложных функций и взаимодействия двоих студентов при решении задачи по электронике. Результаты экспериментов приведены в статье [13]. На кафедре МПЧС проводились исследования изменения энергоинформационных потоков студентов в процессе прослушивания лекции и их реакция на дистанционное телепатическое воздействие из другой аудитории. Студенты выбирались для участия из различных групп успеваемости. Результаты экспериментов приведены в статье [14]. Данный метод и прибор используется в учебных программах курсов «Синергетика» и «Компьютерные технологии в образовании».

Заключение

Можно сделать вывод о том, что группа лиц, участвующих в процессе обучения, может рассматриваться как неравновесная термодинамическая система, обладающая свойствами открытости, нелинейности и когерентности. Термодинамические показатели, такие как энтропия и функция диссипации, могут служить количественными характеристиками энергоинформационных потоков при обучении. Посредством оценки сте-

пени хаотичности энергоинформационных потоков в сознании человека можно оценить глубину сосредоточенности обучающегося на изучаемом материале, имплицативную связь между группой лиц, участвующих в процессе обучения, и динамику самого процесса.

Литература

1. Васильев В.Н., Дульнев Г.Н., Стражмейстер И.Б., Агеев И.Л., Чашин А.В. Способ определения локального удельного потока энтропии человека. Патент на изобретение № 2290058 от 27.12.2006.
2. Дульнев Г.Н. В поисках тонкого мира. – СПб: Весь, 2004. – 286 с.
3. Дульнев Г.Н., Ипатов А.П. Исследование явлений энергоинформационного обмена: экспериментальные результаты. – СПб: СПб ГИТМО (ТУ), 1998. – 70 с.
4. Дульнев Г.Н. Энергоинформационный обмен в природе. СПб: СПбГУ ИТМО, 2000. – 135 с.
5. Дульнев Г.Н. Введение в синергетику. – СПб: Проспект, 1998. – 256 с.
6. Волькенштейн М.В. Энтропия и информация. М.: Наука, 1986. – 192 с.
7. Николас Г. Пригожин И. Самоорганизация в неравновесных системах. – М.: Мир, 1979. – 512 с.
8. Потеев М.И. Развитие профессионально-педагогического образования в университетах технического типа. – СПб.: Ива, 2005. – 440 с.
9. Бриллюэн Л. Научная неопределенность и информация. – М.: Иностранная литература, 1968. – 193 с.
10. Музалевская Н. И., Бобров А. В. О возможной роли двойных электрических слоев в реакции биологических объектов на внешнее воздействие // Биофизика. 1988. Т.33. №4. С. 725.
11. Полевые информационные взаимодействия. Сборник трудов. / Под ред. Боброва А.В. Орел: ОрлГУ, 2003. – 568 с.
12. Дульнев Г.Н., Стражмейстер И.Б. Способ оценки степени хаотичности энергоинформационных потоков в сознании человека // Научно-технический вестник СПбГУ ИТМО. 2005. Т.18. С.59–62.
13. Дульнев Г.Н., Стражмейстер И.Б., Богданцев А.С. О возможностях и перспективах исследования процесса обучения человека регистрацией изменения в нем энергоинформационных потоков // Научно-технический вестник СПбГУ ИТМО. 2006..
14. Дульнев Г.Н., Стражмейстер И.Б. Оценка познавательных способностей человека по термодинамическим показателям // Научно-технический вестник СПбГУ ИТМО, 2005. Т.20. С. 3–8.

**ЛОЖНЫЕ ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ
ДЛЯ ИССЛЕДОВАНИЯ И ОТВЛЕЧЕНИЯ АТАК****Н.М. Лукьянов, А.М. Дергачев**

В статье рассматриваются проблемы массового использования ложных вычислительных систем для исследования и отвлечения атак на вычислительную систему. В ней рассматривается возможность применения таких систем совместно с сетевыми экранами и системами обнаружения вторжений. В задачи исследования входила разработка прототипа ложной вычислительной системы и пробная эксплуатация в составе реальной вычислительной сети, а также оценка работоспособности этого прототипа по результатам испытаний.

Введение

В связи с постоянно растущим числом корпоративных сетей, сетевых сервисов и пользователей Интернет самым острым образом встает вопрос о безопасном функционировании информационных систем.

По данным БиБиСи, количество зараженных компьютеров, подключенных к Сети, по состоянию на январь 2007 составляет 25% от общего числа [18]. За последние 6 лет число критических уязвимостей операционных систем и приложений увеличилось в 8 раз и в 2006 году составило 8,064 (данные CERT). Попеременно или день в день выходят и объявления об уязвимости, и вирусы, использующие эту уязвимость. Программы, обеспечивающие злоумышленникам несанкционированный контроль над компьютером, называемые ботами, также вносят вклад в усугубление киберугроз.

Вся защита на сегодняшний день основана на блокировании и принятии контрмер против наиболее распространенных типов атак. Однако при появлении очередного нового метода атаки или ранее не известной уязвимости (так называемой «уязвимости нулевого дня») защищаемая сторона оказывается уязвимой.

Поэтому применяются новые средства мониторинга и активной защиты, такие как система обнаружения вторжения и системы предотвращения вторжения. Однако они также обладают рядом недостатков: они могут фиксировать ложные тревоги и требуют постоянного обновления сигнатур.

Именно такая сложная ситуация с существующими средствами защиты сетей и побуждает искать дополнительные средства защиты и отвлечения атак. Идея создания специальной среды для изучения действий атакующего впервые встречается в публикации Билла Чесвика (Bill Cheswick) «An evening with Berferd», описывающей действия администратора после обнаружения попытки проникновения. В этой ситуации для отслеживания действий злоумышленника была создана замкнутая среда при помощи широко используемых в UNIX технологий chroot и jail. С тех пор такие системы принято называть «honeypot» (от английского «бочонок с медом», «приманка») или ложные вычислительные системы (ЛоВС). В отличие от систем обнаружения и предотвращения вторжений, технология ложных вычислительных систем предлагает заманивать атакующих в систему и анализировать их действия с самого начала. Ложная система представляется атакующему как настоящая. Главная ее задача – отвлечение внимания атакующей стороны от реальной цели и получение как можно более полной информации об атакующих.

На сегодняшний день существует небольшое количество ЛоВС, и они не могут использоваться в масштабе предприятия. У ложных систем существует множество нерешенных проблем. Прежде всего – это сложность настройки и поддержки системы –

разрозненность компонентов, отсутствие готовых к работе систем, затраты аппаратных ресурсов. Также не решена проблема сбора и обработки данных. Результаты работы системы не всегда понятны, поскольку существует несколько источников данных, у каждого из которых – свой собственный интерфейс пользователя.

Для решения этих проблем можно предложить следующие пути. С помощью виртуализации систем можно исключить избыточность оборудования и сократить трудоемкость развертывания систем, разработать специальных программных агентов сбора и обработки данных, а также единый вариант представления данных на основе веб-интерфейса.

Классификация ЛоВС

В общем случае ЛоВС можно классифицировать по степени взаимодействия со злоумышленником. Выделяются два основных вида систем – слабого и сильного взаимодействия.

ЛоВС слабого взаимодействия, как правило, имитируют некоторую часть сетевых сервисов. Злоумышленник ограничен во взаимодействии с этими сервисами. Главная цель ЛоВС слабого взаимодействия – обнаружение сканирований и несанкционированных попыток соединения. ЛоВС может быть просто установлена и настроена в соответствии с требованиями с помощью скриптов или сценариев. Задачи администратора в данном случае – производить мониторинг предупреждений, которые генерирует ЛоВС, а также отслеживать изменения имитируемого сервиса. Такие ЛоВС полезны при массовых атаках роботов, на Интернет-порталах, сетях беспроводной передачи данных, когда массовость обращений к сервису компенсируется его низким уровнем взаимодействия со злоумышленником.

ЛоВС сильного взаимодействия предоставляют большее количество информации о злоумышленниках, но требуют достаточное время для построения и поддержки. Цель ЛоВС сильного взаимодействия – предоставить злоумышленнику доступ к реальной операционной системе, где ничего не имитируется или ограничивается. Существует возможность исследовать новые средства, обнаруживать неизвестные уязвимости в операционных системах или приложениях, а также узнавать, каким образом злоумышленники связываются между собой.

Обычно ЛоВС сильного взаимодействия располагаются в контролируемой среде, например, в сети – после сетевого экрана. Способность контролировать злоумышленника происходит не только от самой ЛоВС, но и от контролирующего сетевого устройства – сетевого экрана. Сетевой экран предоставляет злоумышленнику возможность атаковать ЛоВС, но запрещает производить внешние атаки.

Использование виртуальных машин

В последнее время серверная виртуализация получила широкое распространение, и технологии, такие как VMware, сейчас очень популярны. Виртуализация серверов – это размещение нескольких независимых операционных систем на одном физическом компьютере, эмуляция сетевых интерфейсов и коммуникационного оборудования.

Виртуализация ложных вычислительных систем предполагает выполнение на одном физическом сервере нескольких ОС, каждая из которых может быть использована как для эмуляции отдельных служб, так и в качестве полноценных ложных систем сильного взаимодействия. Предусмотренные образы ложных систем позволяют решить проблему сложности развертывания и обслуживания ЛоВС, что так необходимо при масштабном применении системы.

На рис. 1 можно увидеть, как на одном сервере размещаются три ложные системы – две ложных системы слабого взаимодействия и одна ложная система сильного взаимодействия. Используется полная виртуализация на базе виртуальных машин VMWare.

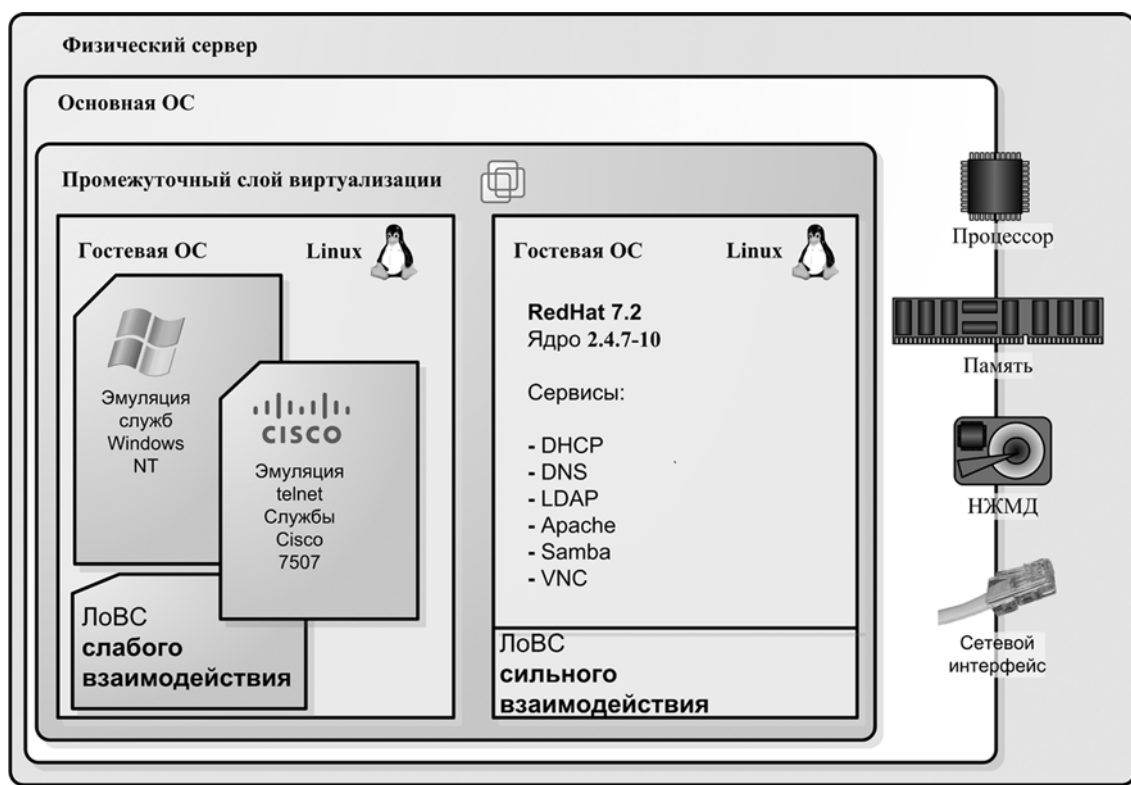


Рис. 1. Принцип полной виртуализации ЛоВС

Гостевые операционные системы, запущенные на основной системе, представляются атакующему как реальные независимые системы. Все пакеты, поступающие на внешний интерфейс основной системы, передаются на гостевые операционные системы – виртуальные ЛоВС. Через свои интерфейсы ЛоВС не имеют доступа к производственной сети и управляющим системам, что обеспечивает необходимый уровень безопасности. В случае взлома гостевой ОС атакующий не сможет использовать такую систему для последующих атак.

Анализ и обработка данных

Для систем слабого взаимодействия сбор данных организуется с помощью службы журналирования syslog, а данные ложных систем сильного взаимодействия отправляют с помощью специального скрытого модуля ядра sebek.

Работа sebek основана на подмене системного вызова read(). Таким образом, эта утилита может записывать все данные, к которым через вызов read() имел доступ атакующий. Например, sebek может контролировать системные отчеты SSH-сессий, восстанавливать файлы, скопированные с помощью SCP, а также перехватывать все пароли, используемые злоумышленниками. Собранные данные скрытно пересылаются с клиента на сервер sebek. Скрытие от хакера исходящих пакетов достигается путем модификации сетевого стека ОС.

Для централизованной регистрации событий ложных систем слабого и среднего взаимодействия необходимо перенаправлять все регистрируемые события из локального хранилища ложной вычислительной системы в центральное хранилище. Для этого необходимо настроить службу журналирования событий syslog таким образом, чтобы

все зарегистрированные события ложной системы отправлялись на удаленную систему регистрации событий.

Программный агент сбора и обработки данных, находящийся на шлюзе ЛоВС, преобразует данные с ложных систем слабого и сильного взаимодействия к единому формату для дальнейшего сопоставления данных из других источников, таких как, например, систем обнаружения вторжений. После этого уже однородные данные могут анализироваться готовыми средствами анализа данных.

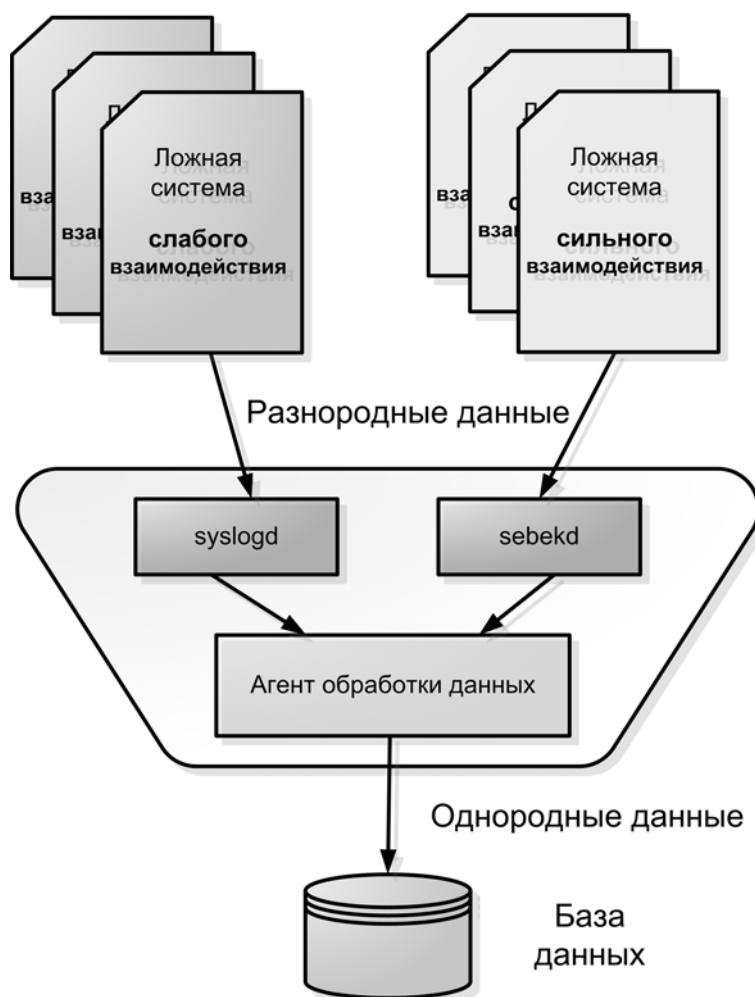


Рис. 2. Сбор и обработка данных

Испытание системы

Для оценки работоспособности и получения практических результатов работы системы на основе рассмотренных выше компонент был создан прототип (см. рис. 3). Он состоял из двух серверов. Первый сервер являлся шлюзом в Интернет, а также собирал данные о работе ложных систем. У него было три сетевых интерфейса.

Все пакеты, поступающие на внешний интерфейс шлюза, обрабатывались системой обнаружения вторжений и передавались на виртуальные ложные системы, расположенные на втором сервере. Через свои интерфейсы ложные системы не имели доступа к производственной сети и управляющим системам. Информация о работе ложных систем поступала и обрабатывалась на сервере обработки данных.

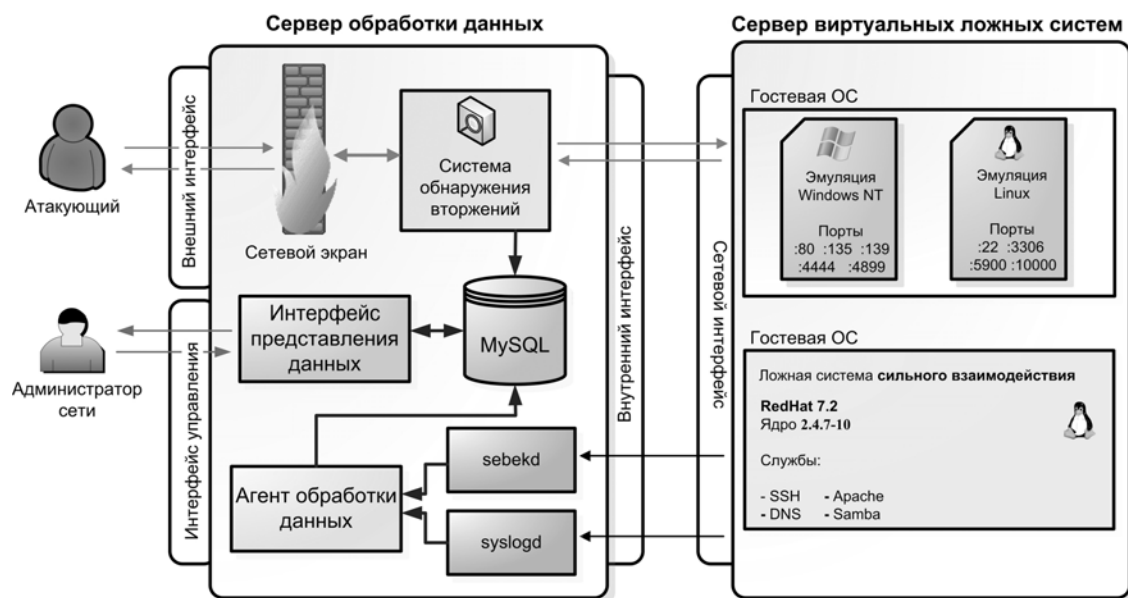


Рис. 3. Прототип системы

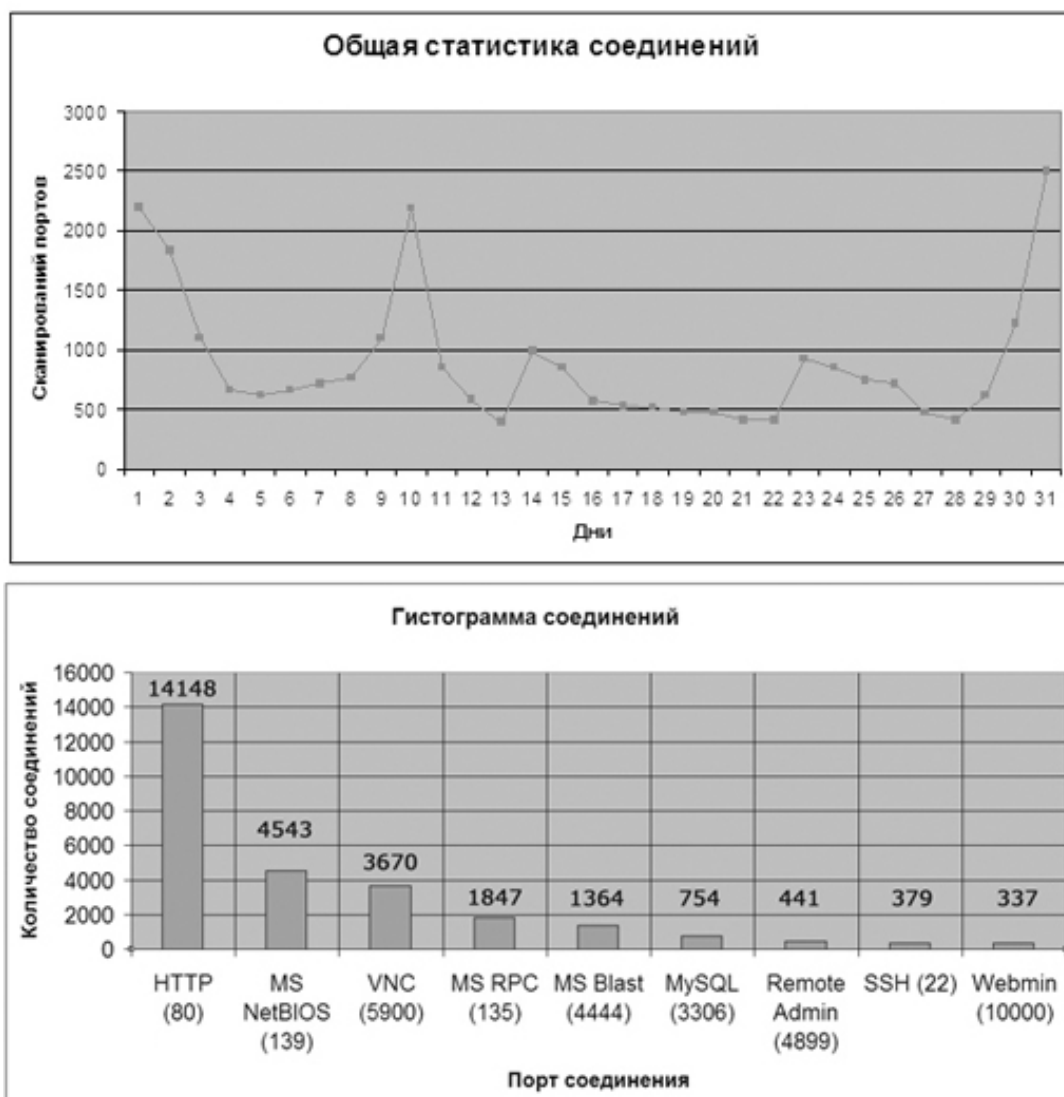


Рис. 4. Статистика соединений

Данный прототип ложных вычислительных систем был размещен в реальной производственной сети. Он состоял из 2 ложных систем слабого взаимодействия, имитирующих серверы на ОС Linux и Windows, и 1 ложной системы сильного взаимодействия на базе систем Linux RedHat.

Испытания системы проводились 1 месяц, в течение которого было зафиксировано 27483 соединений (см. рис. 4). Порт с наибольшим количеством соединений – это порт 80. Соединения на этот порт – преимущественно попытки атаки на веб-сервер Microsoft IIS, который имитирует ложная система слабого взаимодействия. Также очень большое количество соединений производилось по порту 139, атаку на который проводит вирус Sasser.

Данный график также показывает, что ложная система была множество раз атакована вирусом – червем под названием MS Blast, который использует порты 139 и 4444.

Было произведено 3670 попыток соединения со службой удаленного администрирования VNC, в основном в надежде на отсутствие пароля или с попытками его перебора. Меньшее количество соединений у портов служб базы данных MySQL – 754. Со службой удаленного администрирования Windows машин Remote Administrator соединялись 441 раз, также как и в случае с VNC для перебора пароля. Со службами удаленного доступа SSH и веб-службой удаленного администрирования Webmin соединялись 379 и 337 раз соответственно.

Заключение

В результате исследований был предложен и описан целый ряд решений, призванных устранить недостатки существующих ложных вычислительных систем. Результаты исследований и проведенных испытаний подтверждают правильность выбора решений и архитектуры системы.

В построенной системе используется 2 типа ложных систем – слабого и сильного взаимодействия. Системы слабого взаимодействия имитируют отдельные службы, такие системы обычно используются для обнаружения сканирований и попыток несанкционированных соединений. Системы сильного взаимодействия – это настоящие, полностью открытые для взаимодействия операционные системы. Комбинация ложных систем разных типов взаимодействия позволяет быстро и точно копировать поведение реальных систем.

Применение технологии виртуализации дало возможность совместного размещения ложных систем разного уровня взаимодействия на одном и том же физическом сервере, а также позволило оперативно создавать и развертывать ложные системы по имеющемуся шаблону. Виртуализация также стала еще одним препятствием для атакующих в случае взлома ложных систем.

Использование программных агентов сбора и обработки данных и модели данных, не зависящей от источника, позволило преобразовывать данные с ложных систем слабого и сильного взаимодействия к единому формату для дальнейшего сопоставления данных из других источников, таких как, например, систем обнаружения вторжений. Использование схемы с центральной базой данных и возможность полностью импортировать данные из всех ЛоВС позволяет затрачивать гораздо меньше человеческих ресурсов при развертывании и обслуживании большого количества ЛоВС. Использование единого веб-интерфейса пользователя, который отображает данные об атаке с различных источников с максимальной детализацией, облегчает процесс управления и мониторинга системой.

Описанные выше решения основных проблем ЛоВС позволяют использовать эти системы не только в академических целях, но и в масштабах предприятия. И, наконец, самый главный результат – это положительный опыт эксплуатации системы, получен-

ный в результате продолжительных испытаний в реальной вычислительной сети, находящейся в промышленной эксплуатации.

Литература

1. Reto Baumann, Christian Plattner. Honeypots. Swiss Federal Institute of Technologie Zurich, February 2002.
2. Grimes Roger A. Honeypots for Windows // Apress, 2005. ISBN 1590593359
3. Jones R. Honeynets: An Educational Resource for IT Security / Proceedings of ACM SIGITE, Salt Lake City Utah, 2004.
4. HoneyNet Project. Know Your Enemy: Learning about Security Threats (2nd Edition). Addison-Wesley Professional, 2004.
5. Michael O'Leary, Shiva Azadegan, Jay Lakhani. Development of a Honeynet Laboratory: a Case Study. Towson University, January 2006.
6. Джонс М. Т. Виртуальный Linux. Обзор методов виртуализации, архитектур и реализаций // IBM developer Works, 2007.

SDM-МОДЕЛЬ КЛИЕНТ-СЕРВЕРНОЙ СИСТЕМЫ

А.О. Зверев, Б.Д. Тимченко

Рассматривается возможность применения технологии SDM для представления модели архитектуры клиент-серверной системы. В работе анализируется состояние вопроса и особенно состояние инструментальных средств создания моделей приложений. Иллюстрируется применение SDM-технологии на примере восстановления архитектуры фрагмента клиент-серверной системы, реализованной на платформе .Net.

Введение

В архитектуре современных распределенных систем (РИС) [1] повсеместно и в различных формах используется клиент-серверная организация, согласно которой сервер предоставляет клиенту обслуживание, определяемое как сервис. Данная абстракция применима на любом уровне рассмотрения – от нижнего уровня взаимодействия компонент до системы в целом. В соответствии с этим РИС стремятся рассматривать как совокупность IT-сервисов, из которых создается архитектура, обеспечивающая потребности бизнес-процессов и определяемая как сервис-ориентированная архитектура (Service-Oriented Architecture – SOA).

Движение к SOA означает не только отказ от «монолитных» и «жестких» информационных систем, но и дальнейшее существенное усложнение систем как в разработке, так и в управлении их использованием. Управляемость становится важнейшим эксплуатационным качеством современных РИС. Преодоление сложности в управлении IT-сервисами, создание соответствующих инструментальных средств всегда было в центре внимания компаний-разработчиков [2, 3], предлагающих различные решения для управления качеством обслуживания, анализа, управления изменениями. Это направление, определяемое как Information Technology Service Management (ITSM) – управление IT-сервисами, активно развивается в последние годы как методически, так и инструментально. Другое движение в преодолении сложности и обеспечении качества состоит в повсеместном использовании моделей: от моделирования бизнес-процессов до генеративных методов создания приложений на основе моделей. Данное направление – не новое, достаточно вспомнить многочисленные средства моделирования бизнес-процессов (IDEF, BPL, Windows Workflow и многие другие) или UML-базируемые решения [4, 5]. Имеются впечатляющие результаты и не менее внушительные сложности. Одной из главных считается проблема слабой связи модельных решений, применяемых на различных этапах жизненного цикла систем, а также недостаточное внимание к проблеме управляемости.

В последнее время активную работу в развитии модельного подхода ведет компания Microsoft, что выразилось в объявлении инициативы DSI (Dynamic Systems Initiative) и масштабной работе по развитию методов и инструментов ее обеспечения. Основой подхода является концепция System Definition Model (SDM) и реализующий ее язык, инструментальные и технологические средства. В совокупности образована новая SDM-технология, впервые представленная в России в конце прошлого года в составе Visual Studio 2005 Team System (VS2005 TS).

Опыт использования технологии SDM еще только нарабатывается, в самой технологии еще неизбежны изменения, однако и в настоящем виде она заслуживает внимания и обобщения опыта. В данной работе представляется реализация SDM-модели одной из компонент клиент-серверного приложения, обладающего многими характерными признаками распределенных систем и разработанного на платформе .Net.

Жизненный цикл приложения

Создание и использование современных распределенных систем требует внимания к модельному обеспечению всех этапов жизненного цикла приложений. В идеале

необходим некий унифицированный подход, поддерживающий не только отдельные этапы, а жизненный цикл приложения в целом. Замысел SDM и его предварительное рассмотрение позволяет говорить как о перспективности технологии в указанном выше отношении, так и о необходимости детального исследования возможностей и выявления ограничений.

Будем рассматривать жизненный цикл приложения состоящим из этапов:

- разработка;
- развертывание;
- функционирование.

Заметим, что в таком же виде можно представить и жизненный цикл системы в целом. Полная переработка, крупные изменения в релизах и постоянные изменения, типичные для РИС, могут вызвать повторения этапов 1 и 2. Рассмотрим модельную и инструментальную обеспеченность отдельных этапов.

Разработка (Development) приложений, несомненно, является этапом, наилучшим образом обеспеченным моделями. В CASE-системе IBM Rational, базирующейся на UML, модельный подход реализован наиболее последовательно. Можно сослаться на многие другие реализации, например, Oracle Designer [6]. Разработка на основе моделей совершенствуется, и по этому вопросу имеется огромное количество публикаций. Полезно заметить, что модели, ориентированные на разработку, в той или иной степени связаны с моделированием бизнес-процессов. Другая деталь состоит в том, что модели этапа разработки пока не отражают свойств инфраструктуры, в которой приложения будут функционировать.

На стадии развертывания (Deployment) происходит в той или иной степени связывание приложения с ресурсами РИС в результате конфигурирования и действий по настройке параметров операционной системы и промежуточного ПО. Конфигурационные данные по различным причинам подвержены изменениям, что влечет за собой трудно предсказуемые изменения в работе приложений (качестве IT-сервисов). Обеспечение контроля за изменениями в методологии ITSM считается сложнейшей задачей и выделяется в отдельную составную часть Configuration and Change Management (Конфигурирование и управление изменениями). Важно заметить, что эта работа выполняется чаще всего без учета архитектуры приложений и модели этапа разработки.

На стадии функционирования (Operations) сервис реализуется, и здесь же проявляется влияние всех факторов, присущих конкретной инфраструктуре. Сервис существует как конкретный системный экземпляр ранее абстрактного приложения. На этой стадии возникают многочисленные проблемы и задачи – оценивание качества обслуживания сервиса, выявление проблем и оптимизация приложений, контроль за изменениями и воздействие на конфигурацию с целью достижения необходимых значений показателей функционирования. Распределенность многократно усложняет эти задачи по сравнению с централизованными системами.

Основной технологией обеспечения этапа функционирования является мониторинг и соответствующая обработка первичных данных для получения показателей функционирования. Это может быть обеспечено только на основе рациональной архитектуры мониторинга, которая, в свою очередь, должна учитывать архитектуру приложения и конфигурацию развернутого экземпляра.

Состояние модельно-инструментального обеспечения

В оценке перспектив использования любой новой технологии необходимо учитывать состояние и использование уже существующих средств. Наличие большого числа инструментов и подходов к рассматриваемым проблемам сильно усложняет получение аналитических оценок, являющихся к тому же в основном чисто качественными. В свя-

зи с этим для анализа нами избраны только наиболее масштабные и устоявшиеся подходы.

Не вызывает сомнения тот факт, что модельные подходы и решения, основывающиеся на UML, должны быть признаны наиболее распространенными. Имеется также значительный опыт их использования, который и дает основания для выводов. Применительно к обеспечению жизненного цикла приложений UML-технологии доказали эффективность на этапе проектирования и разработки. Их эффективность как инструмента бизнес-моделирования не столь очевидна, но это вопрос индивидуальных предпочтений. Что касается последующих этапов жизненного цикла приложений, то ситуация иная. Формально UML обеспечивает этап развертывания, предлагая для этого диаграммы развертывания (Deployment Diagrams), однако они практически являются средством иллюстрации, а не модельным инструментом, и далее никак не затрагивают конфигурирование. Этап функционирования UML-технологии не обеспечивают вообще. Универсальность UML в связи с этим практически проявляется в отношении предметных областей, для которых создаются приложения, а не этапов их жизненного цикла. Активным сторонником UML-технологии является фирма IBM, владеющая ведущим продуктом этого направления – CASE-платформой IBM Rational. Имеются многочисленные реализации технологий других разработчиков, оценка которых выходит за рамки нашей работы.

Этап функционирования всегда был предметом внимания как крупнейших, так и многих иных фирм. Вновь ограничимся лишь наиболее значительными решениями – HP OpenView и IBM Tivoli, решающими такие ключевые задачи, как конфигурирование, управление изменениями, мониторинг/оценивание и управление качеством обслуживания. Это мощные ITSM-ориентированные решения, прошедшие испытания временем. Оставляя подробности, отметим, что в этих решениях этап разработки никак не затрагивается.

Microsoft, позднее других развернувшая работы по обеспечению управляемости РИС, несомненно, учитывает имеющийся опыт. Амбициозное стремление реализовать некий унифицированный подход (этапы жизненного цикла) выражается в призыве создавать приложения с «мыслью» об их функционировании (Design for Operations). По нашему мнению, это дает основания для самого серьезного внимания к SDM-технологии, ее активной апробации и исследованиям в этом направлении.

Эксперимент на примере решения задачи обратного проектирования

SDM-технология [7, 8], реализованная в составе VS2005 TS, прежде всего ориентирована на применение в прямом проектировании и последующей жизни приложений. В этом случае SDM-модели создаются в ходе проектного процесса, естественным образом развиваясь и обеспечивая последовательные этапы жизненного цикла. Такой опыт, однако, ограничен самой новизной подхода, на освоение и накопление опыта нужно время.

Представляется, что общность подхода позволяет искать его применение и вне рамок прямого проектирования, в частности, в задаче обратного проектирования. Будем понимать под ней задачу восстановления архитектуры приложения на уровне выше исходного документированного текста. Практическую значимость этой задачи трудно переоценить: на этапах развертывания и функционирования документированная архитектура приложений часто вообще отсутствует.

Ориентация на сервисное представление, по нашему мнению, может быть достигнута, если модель (представление) архитектуры строить в терминах workflow. Это важно еще и потому, что этот механизм представления процессов реализован в Microsoft Windows Workflow Foundation и интегрирован в VS2005 TS. Ниже представляется экс-

перимент по обратному проектированию с использованием SDM-технологии, в котором исходной является одна из составляющих конкретной клиент-серверной системы, разработанной на платформе .Net без использования моделей.

В качестве конкретного примера восстановления архитектуры системы был взят один из этапов процесса инициализации системы. Этот процесс предполагает создание персистентных объектов всех типов и загрузку в них значений из базы данных. Следует понимать, что эти объекты взаимосвязаны, а, следовательно, имеется строгая последовательность в создании объектов.

На начальном этапе имеется исходный код приложения и набор инструментальных средств, обеспечивающих SDM-технологии. Рассмотрим предложенную последовательность действий решения поставленной задачи.

1. Производится построение диаграммы классов с использованием стандартного средства VS2005 Class Diagram (рис. 1). По диаграмме классов и исходному коду системы можно найти методы, образующие основу сервисов системы, с целью получения знания алгоритмов работы сервисов. Следует отметить, что полученная диаграмма имеет нотацию, отличную от UML, который в составе SDM-технологии не используется.

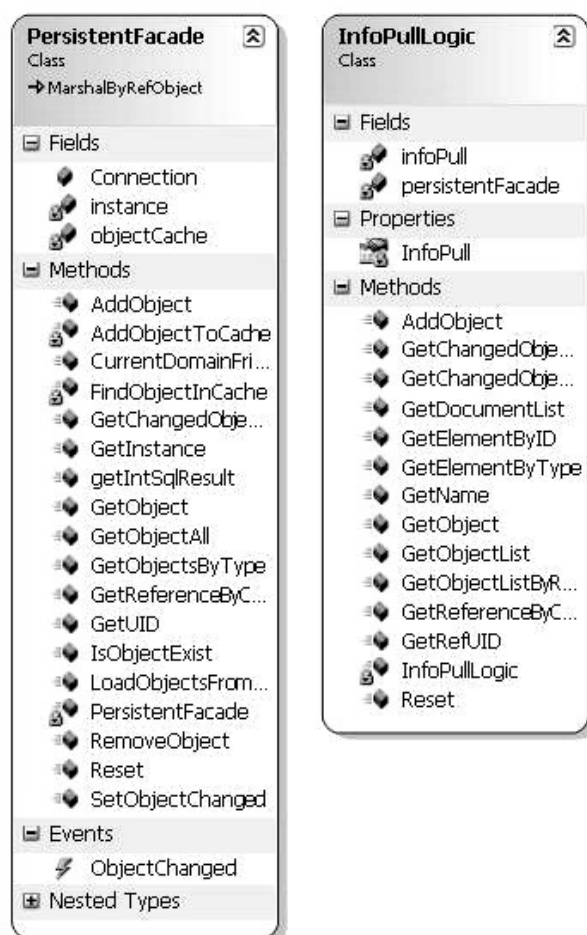


Рис. 1. Диаграмма классов

По полученной диаграмме представляется возможным сделать следующие выводы. В алгоритме задействованы класс логики InfoPullLogic и классы механизмов поддержки персистентных объектов: PersistentFacade, PersistentObject, InfoObjectBase, UID. Как видно из диаграммы классов, эти классы не имеют между собой отношений типа наследования.

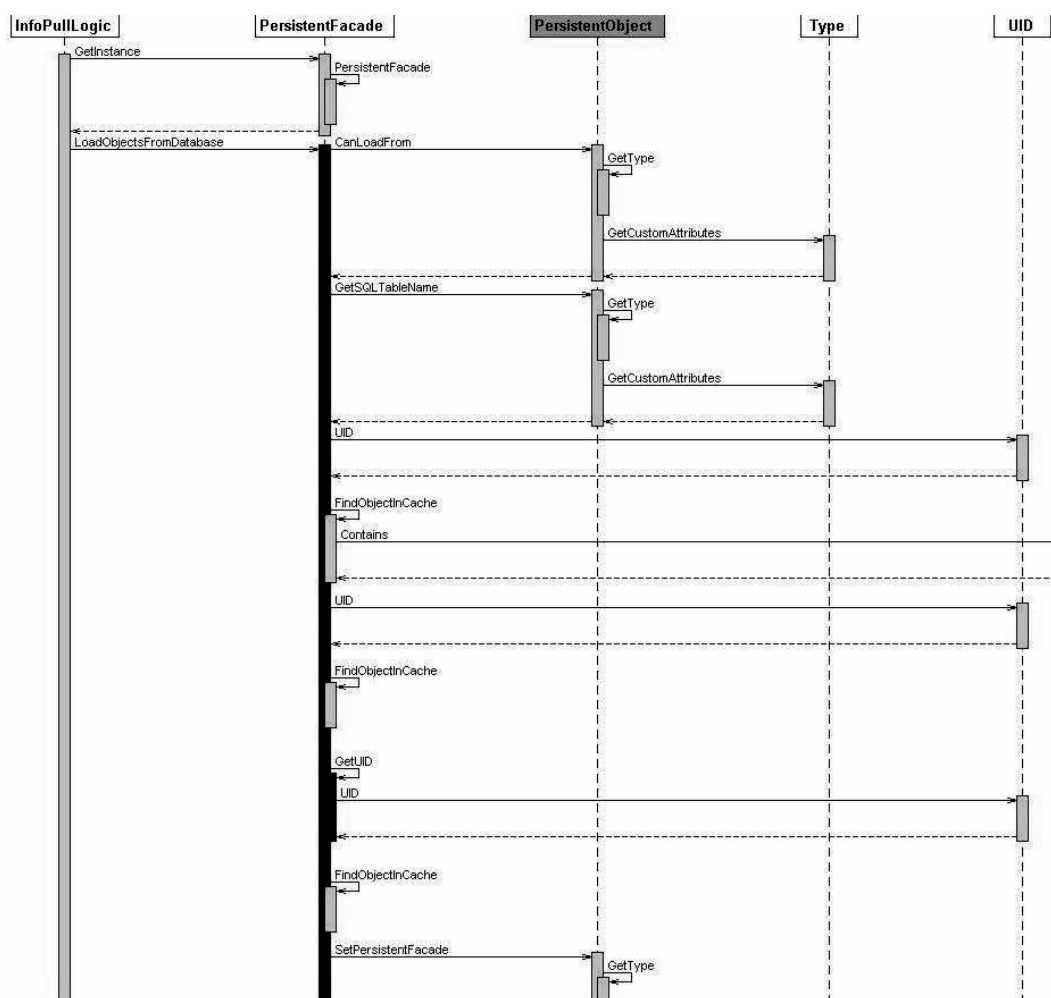


Рис. 2. Диаграмма последовательности вызовов

2. Производится построение диаграммы последовательности вызовов с использованием инструмента CodeLogic (рис. 2). Также при помощи этого инструмента можно построить необходимые представления диаграммы классов и блок-схемы работы метода. Целью является понимание алгоритма работы с использованием визуального представления работы метода. На полученной диаграмме последовательности вызовов видно, что при запуске конструктора класса InfoPullLogic выполняется получение экземпляра объекта PersistentFacade (GetInstance). Далее, используя полученный экземпляр, выполняется последовательная загрузка данных по типам объектов (LoadObjectsFromDatabase).
3. Производится проектирование модели Workflow на основе полученных знаний и моделей на предшествующих этапах. Эта модель может быть разработана на основе двух типов шаблонов – последовательной модели и модели конечных автоматов, из которых нами была выбрана последовательная модель представления. Исходя из описания алгоритма, на основе полученных представлений выбираются два объекта, существующих независимо: InfoPullLogic и PersistentFacade. Изначально запускается конструктор класса InfoPullLogic. Во время его выполнения вызывается метод получения экземпляра класса PersistentFacade. При положительном результате в цикле происходит вызов функции LoadObjectsFromDatabase с передачей в качестве параметра необходимого типа объекта. После успешного завершения цикла процесс загрузки данных считается завершенным.

Полученная WF-модель может иметь графическое и текстовое представления, приведенные на рис. 3, 4, соответственно.

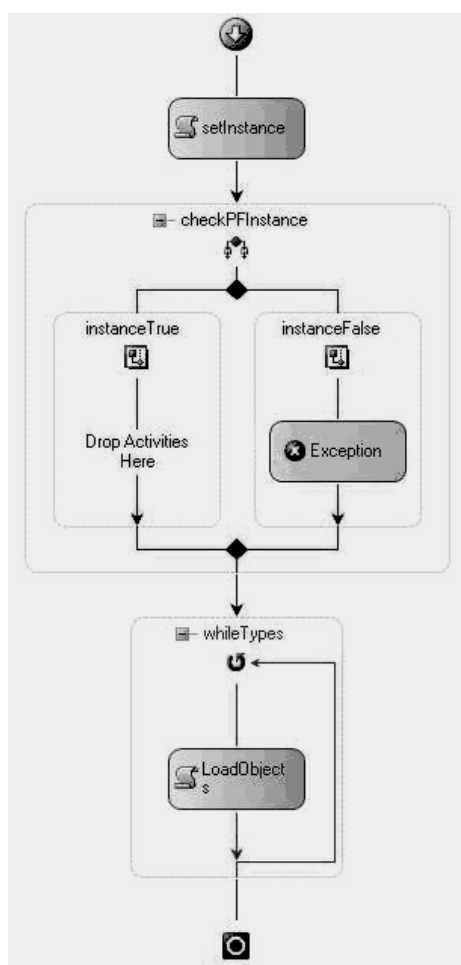


Рис. 3. Workflow модель

```
<SequentialWorkflowActivity
  x:Name="Sequence"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/workflow"
  <CodeActivity x:Name="setInstance" ExecuteCode="setPFInstance" />
  <IfElseActivity x:Name="checkPFInstance">
    <IfElseBranchActivity x:Name="instanceTrue">
      <IfElseBranchActivity.Condition>
        <RuleConditionReference ConditionName="instanceTrue" />
      </IfElseBranchActivity.Condition>
    </IfElseBranchActivity>
    <IfElseBranchActivity x:Name="instanceFalse">
      <IfElseBranchActivity.Condition>
        <RuleConditionReference ConditionName="instanceFalse" />
      </IfElseBranchActivity.Condition>
      <TerminateActivity x:Name="Exception"
        Error="{ActivityBind checkPFInstance,Path=Name}" />
    </IfElseBranchActivity>
  </IfElseActivity>
  <WhileActivity x:Name="whileTypes">
    <WhileActivity.Condition>
      <RuleConditionReference ConditionName="Condition1" />
    </WhileActivity.Condition>
    <CodeActivity x:Name="LoadObjects" ExecuteCode="LoadObjectsFromDataBase" />
  </WhileActivity>
</SequentialWorkflowActivity>
```

Рис. 4. XML-описание Workflow-модели

Заключение

Технология SDM является перспективной как модельное обеспечение этапов жизненного цикла приложения. Хотя в настоящем виде технология реализуема только в среде .Net и, следовательно, только для однородных систем на платформе Microsoft, в будущем положение может измениться. Стандартизация языка, лежащего в основе SDM, происходит с участием ведущих компьютерных фирм, и в настоящее время имеется версия 1.0, получившая название Service Modeling Language (SML). Весьма вероятно, что не только Microsoft, но и другие фирмы будут его использовать. Это открывает перспективу использования подхода и для гетерогенных систем.

Литература

1. Распределенные системы. Принципы и парадигмы. Э. Таненбаум, М. Ван Стеен. – СПб: Питер, 2003.
2. IBM On Demand Workplace <<http://www.ibm.com/ondemand/workplace>>
3. HP OpenView <www.hp.ru/openview/>
4. IBM Rational Software <<http://www-306.ibm.com/software/rational/>>
5. Современные методы и средства проектирования информационных систем. А.М. Вендров. <<http://www.citforum.ru/database/case/>>
6. Oracle Designer <www.oracle.com/technology/products/designer>
7. System Definition Model Overview White Paper
<www.microsoft.com/windowsserversystem/dsi/sdmwp.msp>
8. Overview of System Definition Model (SDM) <[http://msdn2.microsoft.com/en-us/library/ms181772\(VS.80\).aspx](http://msdn2.microsoft.com/en-us/library/ms181772(VS.80).aspx)>

ИСПОЛЬЗОВАНИЕ СПЕЦИАЛИЗИРОВАННОЙ СИСТЕМЫ МОДЕЛИРОВАНИЯ NS-2 ДЛЯ ИССЛЕДОВАНИЯ ТЕЛЕКОММУНИКАЦИОННЫХ СЕТЕЙ

С.В. Храпов

В статье рассматривается проблема моделирования процессов в информационных компьютерных сетях на различных уровнях модели OSI с использованием NS-2.

Введение

Сегодняшний стремительный скачок в развитии телекоммуникационной области однозначно обязан своим появлением компьютерной революции, произошедшей буквально 10 лет назад. Сейчас же мы являемся свидетелями сетевой революции, ибо именно в сети (безусловно, компьютерной) мы находим необходимую информацию, именно в сеть помещаются такие огромные репозитории информации, как Библиотека Конгресса США, библиотека всех печатных работ общества IEEE и т.д. [1]. Не пройдет и десятка лет, как грандиозная часть того, что было создано человеком за всю его историю, будет помещено в сеть. Это лишь пример того, что называется «услуга сети связи». Именно услуги являются той самой силой, которая, с одной стороны, приносит операторам деньги, а с другой стороны – двигает технологию вперед. Если пользователь желает просматривать видеофильмы через сеть, это услуга, которую должен предоставить оператор, а гипотетический производитель оборудования должен, в свою очередь, вкладывать деньги в научно-исследовательскую работу как минимум по следующим направлениям:

- разработка аудио/видео кодеков;
- разработка сетевых протоколов обмена информацией;
- разработка алгоритмов управления перегрузками и т.д.

Мы видим, что услугу с позиции пользователя можно определить как «айсберг», видимую часть которого составляет себестоимость той самой услуги, а огромная подводная часть – затраты производителя и оператора. Известно, что с ростом количества пользователей стоимость услуги резко снижается, т.к. именно пользователям приходится оплачивать весь «айсберг». А насколько велика подводная часть напрямую зависит от используемых методов и технологий оператором и производителем оборудования. В частности, если говорить о сетевых задачах, то необходимость использования компьютерного моделирования разрабатываемой системы очевидна. В нашем примере не последнюю роль в качестве услуги и в формировании ее стоимости будет играть вопрос оптимального распределения ресурсов сети (с точки зрения оператора), а так же используемых технологий (с точки зрения производителя оборудования).

Моделирование динамических систем

Известно, что математическая модель аппроксимирует свойства и поведение исследуемой сети и, как следствие, позволяет решать задачи по оптимизации и ее управлению. Также на модели возможна апробация тех или иных решений, что несравненно дешевле, нежели на реальной системе, и исключает возможные ошибки в ней.

Имитационной является математическая модель, реализованная как программное обеспечение для компьютера и использующая специальные или стандартные языки программирования. При построении подобной модели сети связи могут использоваться как статические, так и динамические модели. При этом под статическими понимаются модели, используемые для исследования состояния сети в заданные моменты времени, например, аналитические методы расчета из теории массового обслуживания, а под динамическими – дискретные стохастические модели, например, процессы генерации

заявок или процессы их обслуживания. Сегодня для решения задач имитационного моделирования сетей связи существует достаточно широкий спектр программных средств – от библиотек функций для стандартных компиляторов до специализированных языков программирования.

Существующее ПО

Сейчас на рынке ПО моделирования сетей связи доступно достаточное количество разноплановых пакетов программ. Если вести речь о многофункциональных пакетах, то среди коммерческих пальму первенства на сегодня занимают следующие:

- COMNET III производства Caci Products Co.;
- BONEs Designer производства Cadence Inc.;
- OPNET Modeler производства Mil3 Inc.

Производитель и название	Цена	Требования к памяти компьютера	Операционные Системы	Примечания
Caci Products Co. COMNET III	\$35.000	от 32 МБ ОЗУ от 100 МБ HDD	Windows 98/NT/2000/XP SunOS, Solaris	LANs, X.25, ATM, Frame Relay, протоколы маршрутизации IP. Реализация собственного кода на SIMSCRIPT. Анимация.
Cadence Inc. BONEs DESIGNER	\$20.000	от 32 МБ ОЗУ от 80 МБ HDD	SunOS, Solaris, HP-UX	LANs, X.25, ATM, Frame Relay, реализация собственного кода на C++. Анимация.
MIL3 Inc. OPNET Modeler	\$40.000	от 16 МБ ОЗУ от 150 МБ HDD	Windows 98/NT/2000/XP, Solaris, HP-UX	Fixed/wireless LANs, X.25, ATM, Frame Relay, Intelligent Networks, Web caching, http и т.п., реализация собственного кода на C++. Анимация. Исходный код библиотек частично открыт.
VINT project Network Simulator version 2 (NS-2)	-	от 8 МБ ОЗУ до 250 МБ HDD	Windows 95/98/ME/NT/2000/XP, Solaris, SunOS, Linux, FreeBSD HP-UX	Fixed/wireless LANs, X.25, ATM, Frame Relay, Web caching, http, все разновидности TCP и т.п., реализация собственного кода на C++ и tcl/otcl. Анимация. Исходный код полностью открыт.

Таблица 1. Сравнение ПО моделирования сетей связи

Все перечисленные программные продукты моделирования, в принципе, обладают практически адекватными наборами возможностей, предоставляемых пользователю. Принципиальное различие заключается только в языке программирования, используемом для реализации пользовательских функций и протоколов. В BONEs и OPNET Modeler в качестве базового языка программирования используется C++, в то время как в COMNET – язык SIMSCRIPT. Самым мощным и, как следствие, самым дорогим и емким с точки зрения ресурсов компьютера среди перечисленных является OPNET Modeler, используемый рядом мировых hi-tech корпораций в своих разработках. Сравнение характеристик упомянутого выше ПО моделирования сетей связи приведено в табл. 1.

Проект NS-2

1996 г. ознаменован началом работ над проектом VINT (Virtual InterNetwork Testbed) [2], организованным DARPA (Defense Research Projects Agency) и реализуемым под руководством целого ряда научных организаций и центров: USC/ISI (University of Southern California / Information Sciences Institute), Xerox PARC, LBNL (Lawrence Berkeley National Laboratory) и UCB (UC Berkley). На сегодня основными спонсорами проекта являются DARPA, NSF и ACIRI (AT&T Center for Internet Research at ICSI).

Главной целью проекта VINT являлось построение программного продукта, позволяющего осуществлять имитационное моделирование сетей связи и обладающего целым рядом характеристик, среди которых – высокая производительность, хорошая масштабируемость, визуализация результатов и гибкость. В качестве основы программной реализации был выбран разрабатываемый в University of California с 1989 г. пакет Network Simulator (до 1995 года известный как REAL). Логично, что для программного продукта было выбрано имя Network Simulator version 2 (далее – NS-2) [3].

NS-2, как и его предшественники, разрабатывался как программное обеспечение с открытым исходным кодом (Open Source code Software - OSS). Такое ПО распространяется бесплатно – без каких-либо ограничений на право использования, модификации и распространения третьими лицами. Таким образом, с точки зрения стоимости NS-2, безусловно, является лидером по сравнению с коммерческим ПО, упомянутым выше. По этой же причине бесплатны и всегда доступны on-line все обновления и дополнения (новые библиотеки, протоколы и т.п.). Еще одним, не менее замечательным свойством программного обеспечения OSS является возможность модификации ядра программы и гибкая настройка в соответствии с требованиями конкретного пользователя. Одним из отличительных свойств NS-2 с точки зрения гибкости является кроссплатформенность. Полные версии, включающие все функции, на данный момент работоспособны под управлением следующих операционных систем:

- SunOS;
- Solaris;
- Linux;
- FreeBSD;
- Windows 95/98/ME/NT/2000/XP.

Для инсталляции полной версии NS-2 необходимо иметь 250 МБ свободного места на диске компьютера и компилятор C++. Существует также упрощенная (компилированная) версия для некоторых ОС, в частности, всех версий Windows, являющаяся не столь гибкой, как полная версия, в частности, невозможно добавлять компоненты, модифицировать ядро и т.п. Однако эта версия очень проста в использовании и не требует глубоких знаний ОС и языка C++. Для функционирования упрощенной версии NS-2 достаточно иметь 3 МБ свободного места на жестком диске компьютера. Но, начиная с версии 2.1b9, полная версия NS-2 может быть скомпонована и ратифицирована под ОС Windows 9x/2000/XP с использованием Cygwin (Cygwin – это Linux-подобная операционная среда под Windows).

Требования к производительности компьютера у NS-2 не столь жестки. В принципе компьютер с процессором 486 может обеспечить приемлемое функционирование даже полной версии NS-2. При необходимости использования NS-2 группой пользователей достаточно иметь инсталлированную полную версию на машине под управлением Unix-подобной ОС. Пользователи могут иметь доступ в режиме терминала к NS-2 и производить необходимые модификации, в том числе и ядра программы, компилируя свою версию в домашнюю директорию. При помощи Nam (Network Animator) возможна анимация полученных результатов.

Архитектура NS-2

NS-2 является объектно-ориентированным ПО, ядро которого реализовано на языке C++. Язык скриптов (сценариев) OTcl (Object oriented Tool Command Language) используется в качестве интерпретатора. NS-2 полностью поддерживает иерархию классов C++ (называемую в терминах NS-2 компилируемой иерархией) и подобную иерархию классов интерпретатора OTcl (называемую интерпретируемой иерархией). Обе иерархии обладают идентичной структурой, т.е. существует однозначное соответствие между классом одной иерархии и таким же классом другой.

Использование двух языков программирования в NS-2 объясняется следующими причинами. С одной стороны, для детального моделирования протоколов необходимо использовать системный язык программирования, обеспечивающий высокую скорость выполнения и способный манипулировать достаточно большими объемами данных. С другой стороны, для удобства пользователя и быстроты реализации и модификации различных сценариев моделирования привлекательнее использовать язык программирования более высокого уровня абстракции. Такой подход является компромиссом между удобством использования и скоростью [4]. В NS-2 в качестве системного языка используется C++, позволяющий обеспечить:

- высокую производительность;
- работу с пакетами потока на низком уровне абстракции модели;
- модификацию ядра NS-2 с целью поддержки новых функций и протоколов.

В качестве языка программирования высокого уровня абстракции используется язык скриптов OTcl, позволяющий обеспечить ряд положительных свойств, присущих языку Tcl/Tk (так как OTcl является объектно-ориентированным расширением языка Tcl):

- простота синтаксиса;
- простота построения сценария моделирования;
- возможность соединения воедино блоков, выполненных на системных языках программирования и простую манипуляцию ими.

```
set ns [new Simulator]

# Создание шести узлов

set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
set n4 [$ns node]
set n5 [$ns node]

# Создание связей между узлами

$ns duplex-link $n0 $n2 2Mb 10ms DropTail
$ns duplex-link $n1 $n2 2Mb 10ms DropTail
$ns simplex-link $n2 $n3 0.3Mb 100ms DropTail
$ns simplex-link $n3 $n2 0.3Mb 100ms DropTail
$ns duplex-link $n3 $n4 0.5Mb 40ms DropTail
$ns duplex-link $n3 $n5 0.5Mb 30ms DropTail

# Установка размера очереди в канале (n2-n3) равного 20

$ns queue-limit $n2 $n3 20
```

Таблица 2. Создание узлов, связей и назначение размера очереди

Объединение для совместного функционирования C++ и OTcl производится при помощи TclCl (Classes Tcl). TclCl является интерфейсом между объектами C++ и OTcl, которым пользуются NS-2 и Nam.

В рассматриваемой архитектуре возможно использование переменных из библиотек C++ объектами OTcl при помощи функции *bind*, в связи с чем переменные C++ могут быть модифицированы через OTcl напрямую. Остальные команды OTcl выполняются путем передачи данных в объект C++ при помощи функции *TclObject::command(int argc, const char*const* argv)*. Для запроса процедуры OTcl из C++ используется класс Tcl. Это может быть полезно, например, при передаче результатов в OTcl из C++. Сценарий моделирования сети связи может быть полностью записан на OTcl, включая параметры линий и узлов, например, задержки, очереди и т.п. Табл. 2 содержит скрипт создания шести узлов и установки связи между узлами.

В случае, если необходимо реализовать какую-либо специфическую функцию, например, дисциплину обслуживания, не реализованную в NS-2 на уровне ядра, используется код на C++. Однако при подобном подходе могут возникнуть неоднозначности. Например, известно, что в NS-2 маршрутизация в основном реализована на OTcl (причем базовый алгоритм Дейкстры – на C++). Если задача состоит в моделировании потоков HTTP, то для работы с каждым потоком достаточно воспользоваться OTcl, при этом обработка на уровне пакетов будет происходить на C++. Реализация однозначно проста. Однако, если количество HTTP потоков в секунду превысит несколько сотен, процедуру работы с потоком лучше перенести в код C++ по причине быстродействия.

Исходная полная версия NS-2 также содержит несколько весьма полезных утилит и средств, например отладчик Tcl, генератор сценариев моделирования и генератор топологии сети. Существует целое множество генераторов топологий, например NTG, RTG, TIRES. В состав же NS-2 входит генератор топологии GT-ITM, при помощи которого можно автоматически создать топологию очень крупной сети без необходимости вручную определять все компоненты сети. Генератор сценариев моделирования, как правило, используется для создания трафика между узлами сети. При моделировании беспроводных сетей этот тип генератора также может быть использован для определения передвижения узлов.

Протоколы

В NS-2 на уровне ядра реализованы почти все известные протоколы сетей связи. Среди наиболее актуальных на данный момент можно отметить следующие: MPLS, IPv6, OSPF, RSVP, протоколы беспроводной связи, web caching и многие другие. Реализовано целое семейство дисциплин обслуживания очередей – RED, WFQ, CBQ, SFQ и т.д. Реализация собственного протокола в рамках NS-2 несложна, если пользователь знаком с принципами построения NS-2, изложенными выше, и с основами программирования протоколов.

К сожалению, весьма слабым местом NS-2 является документация, хотя руководство пользователя существует и всегда доступно в форматах .html и .ps. Это руководство не является исчерпывающим, и некоторые свойства NS-2 в нем не отражены. Поэтому дополнительно к нему рекомендуется использовать другие ресурсы.

Математическая поддержка

Обеспечение сильной математической поддержки позволяет создавать различные виды трафика, начиная от простейшего, подчиняющегося пуассоновскому закону, заканчивая самоподобным. Гибкость архитектуры NS-2 позволяет пользователю реализовывать собственные математические функции на C++.

Модель ошибок

Реализация модели ошибок позволяет моделировать возникновение ошибок на канальном уровне, т.е. искажение или потерю информации. Моделирование возникновения ошибки возможно на уровне битов, пакетов или вероятности в секунду. Возможна также реализация пользовательской модели ошибок.

Визуализация данных

Полная и упрощенная версии NS-2 содержат средство анимации результатов моделирования Nam (Network Animator), реализованное на Tcl/Tk, которое предоставляет графическое воспроизведение проведенного эксперимента – отображение топологии и сети, анимация пакетов, узлов, очередей, различные возможности анализа данных. В качестве входных данных для Nam используются файлы, записанные в процессе функционирования NS-2, т.е. моделирования сети связи.

Заключение

Очевидно, что в настоящее время программный продукт NS-2 является оптимальным средством моделирования телекоммуникационных сетей, по крайней мере, в рамках высшей школы. Проведение экспериментов на базе NS-2 и изучение его архитектуры влечет за собой получение знаний в ряде областей (программирование, телекоммуникации, математика), без которых невозможно существование высококвалифицированного IT-специалиста. На базе NS-2/Nam возможна организация наглядной демонстрации функционирования протоколов и сетевых механизмов, например, влияния дисциплины обслуживания очереди на вероятность потери пакета трафика с разными приоритетами, или сущность различия алгоритмов протокола TCP (slowstart, sliding window, SACK и т.д.).

Весомая часть исследований в области сетей связи в университетах Европы, Азии и США строится на базе NS-2. Почти в каждом университете, готовящем специалистов в области сетей связи, существуют курсы, посвященные NS-2. NS-2 также широко используется и в научно-исследовательских институтах, а в Интернете доступен репозиторий скриптов NS-2 [5, 6].

К сожалению, как показал поиск по ключевым словам с использованием ряда известных поисковых интернет-машин, на территории России NS-2 не используется, или, по крайней мере, используется не широко. Необходимость создания русскоязычной документации на данный момент просто очевидна, а внедрение NS-2 в учебный и исследовательский процессы на территории постсоветского пространства необходимо.

Литература

1. Freericks C. Open Source Standards on Software Process: A Practical Application // IEEE Communication Magazine, April 2001, pp.116–123.
2. Kevin Fall, Kannan Varadhan, and the VINT project, «The ns manual», 2001
3. S. McCanne and S. Floyd, ns Network Simulator. <http://www.isi.edu/nsnam/ns/>
4. Кучерявый Е.А. NS2 как универсальное средство имитационного моделирования сетей связи / Tampere University of Technology, Труды VII международной конференции «Информационные сети, системы и технологии». Минск, 2001.
5. Парамонов А.И. Имитационное моделирование систем и сетей связи. ЛОНИИС, 2000
6. Парамонов А.И. Моделирование сетей связи. ЛОНИИС, 2000.

ФОРМИРОВАНИЕ СВЯЗАННЫХ ТОПОЛОГИЙ СПЕЦИАЛИЗИРОВАННЫХ СЕТЕЙ BLUETOOTH SCATTERNETS

С.В. Храпов

Статья рассматривает проблему формирования связанных топологий в распределенных сетях, построенных по технологии Bluetooth. Обеспечение связности в случаях, когда это выполнимо, является наиглавнейшим требованием для любой системы, нацеленной на поддержку соединений устройств друг с другом.

Введение

Подобно персональным компьютерам предынтернетной эры, распространяющиеся устройства (сотовые телефоны, PDA, карманные ежедневники, и т.д.) сегодня используются прежде всего как автономные, несетевые объекты. Некоторые коммуникационные устройства, такие как сотовые телефоны и пейджеры, соединены с сетью их поставщика услуг, но главным образом устройства не взаимодействуют друг с другом на основе peer-to-peer. В некоторых случаях можно соединить устройства вместе, используя кабельный шнур (например, сотовый телефон к ПК), но этот стиль привязного взаимодействия не только громоздок, но и не может использоваться для того, чтобы соединить более чем два устройства вместе.

Используя Bluetooth интерфейсы, группу устройств в пределах непосредственной близости можно связать и сформировать специализированную сеть, известную как scatternet. Совокупность узлов Bluetooth может быть организована в scatternet многими различными способами, так как Bluetooth основан на расширении спектра, и узлы близости не могут общаться, пока они совместно не используют один и тот же код расширения. В этом состоит отличие от классических узкополосных пакетных сетей радиосвязи или беспроводных LAN систем, где топология взаимосвязей узлов однозначно определена расстоянием между узлами [1].

Понимание топологической структуры распределенных сетей (scatternets) – важный первый шаг к цели создания алгоритмов для *self-organizing networks* (самоорганизующихся сетей).

Меньше чем через пару лет будет возможным строить Bluetooth scatternets, состоящие из десяти и более устройств. Умные офисы будущего, например, будут переполнены всеми видами интеллектуальных устройств – компьютерами, PDA, пейджерами, телефонами, «белыми досками» и датчиками. Необходимы предпосылки для поддержания любого пространства умных приложений взаимодействующих устройств в сети. Следовательно, понимание пространства всех доступных топологий взаимодействия и определение их количества – важный первый шаг по направлению к созданию технологий самоорганизующихся сетей. К сожалению, пространство всех топологий scatternet настолько велико, что в вычислительном отношении недопустимо искать пространство без должного понимания его математической структуры.

Описание

Основополагающим принципом систем Bluetooth является использование метода расширения спектра при скачкообразном изменении частоты (Frequency Hop Spread Spectrum, FHSS). Подобно любой другой системе со скачкообразным изменением частоты, когда два устройства Bluetooth хотят взаимодействовать, они должны сначала «договориться» об общей последовательности скачкообразного изменения частоты. В Bluetooth это достигается на основе согласования роли – основной (master) или подчиненный (slave), которую каждое устройство выполняет. Каждое устройство Bluetooth способно выступать в роли как основного, так и подчиненного. Когда пара устройств

взаимодействует, одно из них действует как основное, другое как подчиненное. Основное устройство выбирает последовательность скачкообразного изменения частоты, а подчиненное устройство следует этой последовательности в течение длительности соединения. После того как соединение установлено, основной и подчиненный узлы совместно используют дуплексный протокол разделения времени (Time Division Duplex, TDD) частотного канала. Основной узел снова ответственен за управление распределением слотов времени. Обычно альтернативные слоты зарезервированы для направлений соединения «основное к подчиненному» и «подчиненный к основному».

Чтобы облегчить взаимодействие между группой устройств, Bluetooth поддерживает формирование того, что называется *пикосетью* (*piconet*). Пикосеть состоит точно из одного основного узла и до семи подчиненных узлов, которые синхронизированы с последовательностью скачкообразного изменения частоты основного узла. Все пары «основной–подчиненный» могут взаимодействовать друг с другом, но подчиненным не разрешается взаимодействовать непосредственно, так как все подчиненные только слушают в течение временного интервала передачи от основных к подчиненному. Нехватка возможностей взаимодействовать на физическом уровне, однако, не препятствует двум подчиненным узлам взаимодействовать. Используя протокол пересылки более высокого уровня, основной узел может всегда коммутировать пакеты среди всех соединенных подчиненных узлов.

Так как каждая пикосеть имеет различную схему скачкообразного изменения частоты, множество пикосетей могут сосуществовать в пространственной близости, не прикасаясь друг с другом. Рис. 1 иллюстрирует пример трех сосуществующих пикосетей [2].

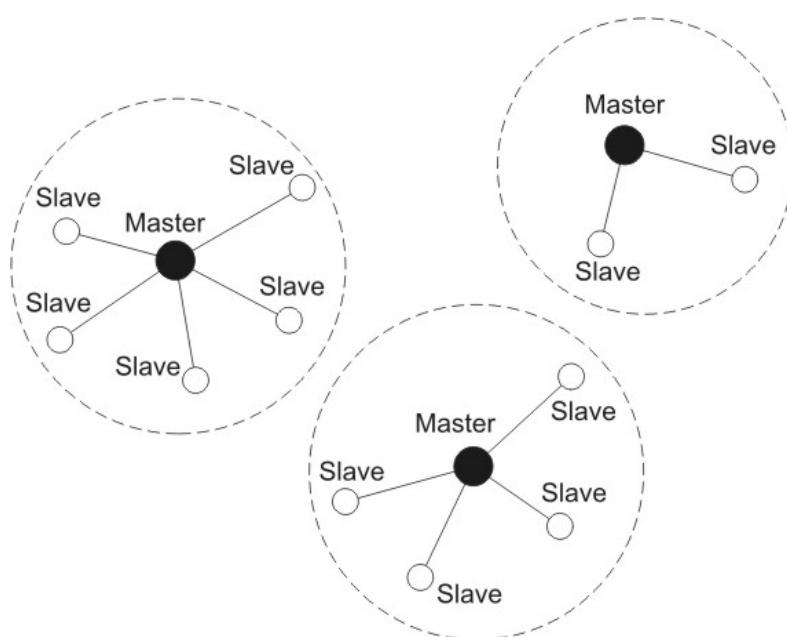


Рис. 1. Пикосети Bluetooth

Чтобы облегчить взаимодействие между пикосетями, Bluetooth определяет понятие *scatternet* (англ. *scatter* – *распределять*, *net* – *сеть*, *распределенная сеть*). Scatternet формируется путем соединения множества пикосетей, используя мосты. Узел, действующий как мост, – это член двух или более пикосетей, и он участвует в каждой пикосети на основе режима разделения времени. После пребывания в пикосети некоторое время узел может обратиться к другой пикосети, переключив последовательность скачкообразного изменения частоты. Периодически изменяя членство во всех пикосетях, узел может посылать и получать пакеты в каждой пикосети, а также пересылать пакеты между пикосетями. Рис. 2 показывает пример scatternet, который сформирован объеди-

нением непересекающихся пикосетей из предыдущего примера. Обратим внимание, что все узлы, действующее как мосты, являются подчиненными. Хотя можно сформировать такие топологии scatternet, где узел действует как основной в одной пикосети и как подчиненный в других пикосетях; такие конфигурации приводят к недостаточному использованию полосы пропускания, и поэтому они нежелательны.

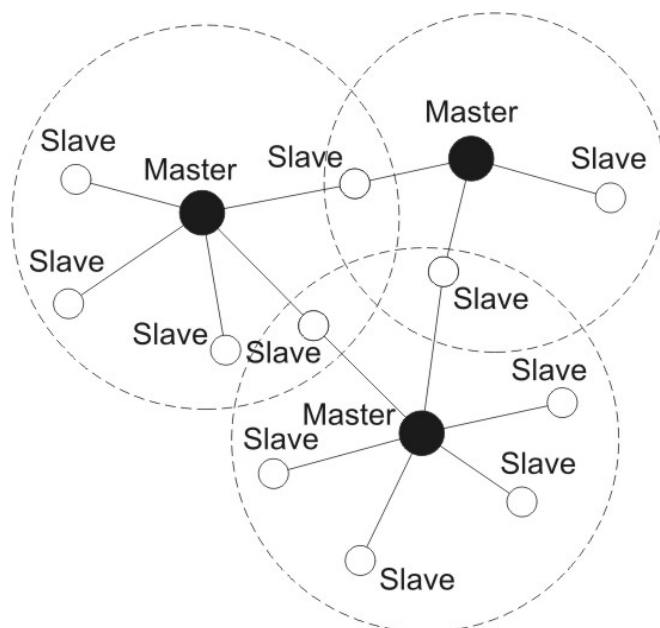


Рис. 2. Распределенная сеть Bluetooth Scatternet

Цели формирования топологий

Цель алгоритмов должна заключаться в генерировании связанных топологий, когда выполнимо удовлетворяющее ограничение степени основного узла, т.е. значение степени, равное семи, самое большее. В этом контексте алгоритмы будут стремиться отвечать на вопросы «Как узлы выбирают их роли?» и «Какой подчиненный узел должен присоединить пикосеть?», при условии, что ответ на вопрос «Сколько подчиненных узлов должен принять основной узел?» налагает спецификация Bluetooth.

Формирование связанной топологии – разумная цель, но существует много случаев, где это, возможно, невыполнимая цель. Например, если существуют две такие группы узлов, что никакой узел в одной группе не находится в диапазоне передачи никакого узла другой группы, то топология будет разъединена независимо от того, какой алгоритм каждый использует. Поэтому важно ясно сформулировать цели алгоритма в таких случаях. Если глобальная возможность соединения будет невыполнима, то логическая топология будет состоять из таких «изолированных участков» (островов) узлов, что каждый остров включает в себя узлы, которые находятся в пределах диапазона передачи друг друга, но узлы в различных островах не могут общаться. В этом случае цель состоит в том, чтобы минимизировать число островов в конечной топологии.

Сетевая модель и проблема сложности

В качестве первого шага по направлению к систематическому исследованию проблемы связанности сформулируем математическую модель системных требований и ограничений. Заметим, что может существовать два типа коммуникационных каналов между любыми двумя узлами. Первый – это физический канал, который существует между любой парой узлов, которые находятся в радиусе связи друг с другом. Другой –

это логический канал Bluetooth, который существует в том случае, когда установлена Bluetooth топология соединений для этой пары узлов. Такое случается, когда один узел является основным, а другой – подчиненным в одной и той же пикосети. Ясно, что условием существования логического соединения между двумя узлами является наличие физического соединения между ними. Сценарий может быть промоделирован с помощью двух графов – графа физической топологии и графа логической топологии. Вершины в обоих графах представляют собой узлы действительной сети. Граф логической топологии, таким образом, является подграфом графа физической топологии. Граф физической топологии задан, а граф логической топологии является результатом работы алгоритма формирования топологии.

Граф логической топологии должен иметь определенные свойства. Согласно спецификации Bluetooth, вершины графа, которые будут назначены на роль основных узлов, могут иметь максимальную степень 7. Для узлов графа, которые будут назначены подчиненными, желательно, чтобы их степени были как можно меньше. В логическом графе подчиненный узел имеет степень, равную единице (канал связи с основным), кроме узлов-мостов, которые имеют степень, равную числу пикосетей, в которых они участвуют. По причине того, что узел-мост со степенью 7, т.е. участвующий в семи пикосетях, можно полагать самым узким местом в системе (он будет обеспечивать наиболее слабую связанность между всеми пикосетями, в которых участвует), как следствие, будем считать 7 ограничением степени узла-моста. Также, хотя это и не является необходимым требованием, желательно, чтобы логический граф был двудольным графом, поскольку множество вершин может быть легко разбито на основные и подчиненные узлы и логические ребра, соединяющие основные и подчиненные узлы и наоборот.

Выполнение обеспечения связи и формирование связанного графа логической топологии, который удовлетворяет желаемым ограничениям – NP-трудная задача для общей физической топологии.

Алгоритмы формирования топологий

Приближаемся к проблеме связанности согласно упрощающим допущениям.

1. Узлы составляют точки в двумерной плоскости.
2. Все узлы имеют один и тот же диапазон передачи.

Обеспечение связи выполнимо тогда и только тогда, когда граф физической топологии связан. Минимальное взвешенное остовное дерево (Minimum Spanning Tree, MST) в графе физической топологии с весом ребра, равняющегося евклидовому расстоянию между узлами, является связанным графом логической топологии [3], который удовлетворяет всем принятым ограничениям.

Построение минимального взвешенного остовного дерева в графе физической топологии обеспечит граф логической топологии, который связан, если связываемость возможна, и состоит из минимального числа компонентов, если связываемость невыполнима. Интересно заметить, что централизованная конструкция алгоритма минимального взвешенного остовного дерева имеет сложность исключительно $O(E \log V)$, если граф физической топологии имеет E связей и V узлов (*Временная сложность распределяющего алгоритма MST – $O(V \log V)$ и стоимость связи – $O(V \log V + E)$ сообщений*), тогда как конструкция графа логической топологии NP-трудная в общем случае.

Исследуем топологии, сгенерированные тремя различными алгоритмами [4, 5].

- Алгоритм MST, который дает гарантии на максимальную степень.
- Поиск в глубину (Depth First Search, DFS), основанный на алгоритме формирования топологии остовного дерева.
- Поиск в ширину (Breadth First Search, BFS), основанный на алгоритме формирования топологии остовного дерева.

DFS (BFS) основывается на алгоритмах формирования DFS (BFS) остовных деревьев, и впоследствии максимальной степенью сегмента выбирается как множество основных узлов. Оба алгоритма удовлетворяют распределенному и динамическому функционированию. Однако ни один не обеспечивает никакой аналитической гарантии на максимальную степень узла [6].

Чтобы оценить работу этих алгоритмов, рассмотрим сети двух типов в наших экспериментах. Первый тип состоит из узлов, однородно распределенных в квадрате размером 1 единица. Второй тип состоит из «группированной топологии» – трех отдельных групп узлов. Позиции центров групп выбраны случайно в квадрате размером 1 единица. Узел может принадлежать одному из трех групп или, возможно, не принадлежит ни одному из групп вообще. Эти 4 события взяты, чтобы быть равновероятными. Если узел принадлежит группе, это соответствует однородному распределению в квадрате группы (стороной 0.4) вокруг центра группы, иначе его позиция однородно распределена в оригинальном квадрате размером 1 единица. Для каждого из этих двух типов сетей оцениваем работу алгоритмов с различным числом узлов (25, 50, 100) и двух различных радиусов передачи (0.4 и 0.6 единиц).

Моделирование сетей и реализация алгоритмов формирования связанных топологий сети были выполнены в специализированной системе моделирования NS-2.

Алгоритм MST генерирует топологии, где основные узлы и мосты имеют низкие максимальные степени. Максимальная степень мостов – между 2 и 3, что существенно ниже, чем аналитическая гарантия, равная 6 для всех узлов. Как ожидалось, максимальная степень и основных узлов и мостов высока в топологии, сгенерированной BFS. Это происходит потому, что основной этап формирования алгоритма приспособлен к генерации «низкорослых» деревьев, которые явно вносят свой вклад в генерацию деревьев очень высокой степени. Таким образом, алгоритм, основанный на BFS, не может использоваться «как есть».

Алгоритм DFS по аналогии имеет основной этап формирования, который сохраняет степени узла минимальными максимально долго, поэтому его работа будет намного лучше [7]. Действительно, он фактически достигает немного более низких степеней, чем MST, и для основных, и для подчиненных узлов, даже притом, что он не обеспечивает никакой аналитической гарантии. Таким образом, можно легко создать примеры, в которых DFS не в состоянии обеспечить низкую степень, охватывающую дерево, предполагая, что в самых практических сценариях степень дерева, произведенного DFS, будет ниже, чем при алгоритме MST [8]. Другое преимущество использования алгоритма DFS состоит в том, что нет никакой потребности в выполнении управлением мощности узлов, чтобы вычислить (или ранжировать) евклидовы расстояния между взаимодействующими узлами. Степень в случае BFS весьма зависит от числа узлов и от радиуса передачи. Напротив, степени, полученные алгоритмами MST и DFS, устойчивы к выбору этих параметров.

Заключение

Важный аспект в контексте распределенных сетей (scatternets) – динамическое функционирование. Динамическое функционирование имеет два главных направления: добавление и удаление узлов, поскольку пользователи прибывают и уходят, и возможны модификации топологии (связности графа), вызываемые мобильными узлами. Проектирование алгоритмов и протоколов, способных к эффективному динамическому функционированию, само по себе является целой исследовательской областью, в рамках которой значимые решения, как правило, должны тщательно объяснять операционные ограничения.

Алгоритмы, представленные в статье, ясно нацелены на поддержку динамического функционирования, полагающуюся только на местную информацию и оптимизирующую обеспечение связи. Однако есть много аспектов динамического функционирования, которые в данной работе не исследовались. Формулировка полного решения для динамической среды – тема дальнейшей работы. Можно констатировать, что результаты этой статьи обеспечат основные конструктивные блоки для разработки такого решения.

Литература

1. D.A. Ghose, A. Razdan, H. Saran, and R. Shorey. Enhancing performance of asynchronous data traffic over the Bluetooth wireless ad-hoc network. // Proceedings of INFOCOM'2001, Anchorage, AK, April 2001.
2. Law, A. K. Mehta, and K.-Y. Siu. Performance of a new Bluetooth scatternet formation protocol. // In Proceedings of MobiHoc'01, Long Beach, CA, October 2001.
3. F. Harary. Graph Theory. Addison-Wesley, 1969.
4. M.A. Marsan, C.F. Chiasserini, A. Nucci, G. Carello, and L. De Giovanni. Optimizing the topology of Bluetooth wireless personal area networks. // Proceedings of INFOCOM'2002, New York, NY, July 2002.
5. N. Johansson, U. Korner, and L. Tassiulas. A distributed scheduling algorithm for a Bluetooth scatternet. // Proceedings of ITC'2001, Salvador, Brazil, december 2001.
6. R. Guerin, E. Kim, and S. Sarkar. Bluetooth technology: Key challenges and initial research. // Conference on Network and Distributed Simulations, 2002.
7. R. Guerin, S. Sarkar, and E. Vergetis. Forming connected topologies in bluetooth ad-hoc networks. University of Pennsylvania, Technical Report, 2002.
8. S. Basagni, G. Zaruba, and I. Chlamtac. Bluetrees-scatternet formation to enable bluetooth-based ad hoc networks. // ICC, 2001.

ВЫЧИСЛИТЕЛЬНЫЕ МЕХАНИЗМЫ КАК ИНСТРУМЕНТ
ПРОЕКТИРОВАНИЯ ВСТРОЕННЫХ СИСТЕМ

А.Н. Лукичев

В статье рассматриваются современные методики проектирования встроенных систем и понятия *паттерна проектирования* и *вычислительного механизма* в их контексте. Предлагаются способы повышения эффективности и ценности применения паттернов. Предлагается трактовка понятия вычислительного механизма и способ формализации вычислительных механизмов.

Введение

За последние несколько лет в области встроенных систем (ВсС) существенно возрос интерес к различным подходам и методам высокоуровневого проектирования. Предлагаемые в результате исследований методики призваны решить такие проблемы, как стремительно возрастающая сложность ВсС, их распределенность, расширение номенклатуры элементной базы (в самом широком смысле – от проводников и дискретных электронных компонент печатных плат до технологий программирования и отладки и блоков IP/VC систем на кристалле), рост надежности и работоспособности [2]. Такие методики, в основном, имеют следующие тенденции:

- повышение уровня абстракции, на котором осуществляется основной объем проектирования системы;
- повторное использование компонент;
- унификация внешних интерфейсов компонент, разработка масштабируемых архитектур;
- использование различных абстракций, таких как «модель вычислений», позволяющих формализовать не только разработку отдельных компонент ВсС, но и способы их организации в систему.

Высокоуровневое проектирование требует наличия средств, позволяющих адекватно отражать и фиксировать определенные свойства системы на верхних уровнях абстракции. Одним из таких средств является паттерн проектирования [4] как способ решения некоторой технической задачи. Паттерны проектирования успешно применяются в объектно-ориентированном подходе к созданию прикладного ПО в течение как минимум последних 10 лет. Существуют попытки применения паттернов в области встроенных систем.

До сих пор, несмотря на все более частые заявления о необходимости абстрагироваться в проектировании ВсС от закрепления за компонентом системы статуса «программный» или «аппаратный», существует во многом искусственное разделение на «отрасли» разработки аппаратуры и программного обеспечения встроенных систем. В работе указывается на принципиальную близость аппаратных и программных решений и наличие общих механизмов, лежащих в их основе.

В рамках аспектной технологии проектирования ВсС [2] предлагается абстракция «вычислительный механизм». Понятие вычислительного механизма четко до сих пор, однако, не определено. В работе делается попытка конкретизировать это понятие. Указывается на сходство понятий «паттерн проектирования» и «вычислительный механизм» в контексте поведенческого аспекта ВсС.

Формализация паттернов проектирования в вычислительной технике в настоящее время в области ООП ограничена применением нотации UML, а в области ВсС зачас-

тую просто отсутствует. В лучшем случае применяются схематические структурные изображения (такие как принципиальная электрическая схема для иллюстрации аппаратного паттерна [1]). В работе обсуждается способ решения этой проблемы для вычислительных механизмов.

Паттерны проектирования

В области ВcC, как и в любой другой области вычислительной техники, не обходится без программирования интерактивных приложений, базирующихся на платформе общего назначения (ПК, пользовательские терминалы, PDA и т.п.). В рамках объектно-ориентированного подхода (ООП) к проектированию такого рода прикладного ПО в последние несколько лет распространилась технология применения так называемых *паттернов проектирования* [4]. С точки зрения ООП, паттерн представляет собой «описание взаимодействия объектов и классов, адаптированных для решения общей задачи проектирования в конкретном контексте» [4]. Таким образом, паттерн представляет собой некий шаблон, по которому строится решение задачи. Задача проектирования при этом обладает в достаточной степени типичностью, т.е. может возникнуть при проектировании разных систем и в разное время. В то же время эта задача существует в контексте, ограничивающем возможность ее возникновения и, следовательно, возможность применения паттерна для ее решения. Согласно К. Александру, впервые употребившему понятие «паттерн» применительно к строительству зданий, «любой паттерн описывает задачу, которая снова и снова возникает в нашей работе, а также принцип ее решения, причем таким образом, что это решение можно потом использовать миллион раз, ничего не изобретая заново» [4]. Паттерн, таким образом, заключает в себе положительный опыт решения какой-либо задачи в процессе проектирования, которая потенциально может возникнуть при проектировании других систем.

В области ООП паттерны получили широкое распространение. В настоящее время существуют обширные каталоги паттернов («языки паттернов»), знание и умение применять наиболее часто используемые из них является требованием, характеризующим базовую подготовку специалиста при приеме на работу, ежегодно проводятся международные конференции PLoP (Pattern Languages of Programs).

Несмотря на большую популярность паттернов проектирования в ООП, а может быть, и вследствие нее, работы по созданию паттернов в проектировании встроенного программного обеспечения ведутся многими группами исследователей, причем не только в рамках объектно-ориентированного подхода. Во многих случаях, однако, это попытки адаптации существующих паттернов ООП. Такие попытки также применяются и по отношению к проектированию аппаратуры, в частности, компонентов «систем на кристалле» (SoC). Дамашевичус и коллеги [3] дают хороший обзор попыткам применения объектно-ориентированных методик к разработке аппаратного обеспечения.

Многие работы в области разработки аппаратных компонент описывают способы решения задач в стиле, сходном с описанием паттерна проектирования [4]: описывается задача или класс задач, контекст, в котором она существует, способ ее решения, а также примеры применения описываемой технологии или методики. Например, Кишиневский и коллеги [5] приводят методику решения задачи взаимодействия компонент в схеме с потенциально изменяемыми задержками передачи сигналов. Описывается контекст, задача («мотивация»), способ ее решения, а также обсуждаются достоинства, недостатки и перспективы, приводятся примеры применения.

Кроме того, существуют попытки систематизации паттернов проектирования встроенных систем [4] и разработки каталогов («языков паттернов»). В таких каталогах собраны паттерны, решающие задачи разработки как программного, так и аппаратного обеспечения ВcC.

Сложность современных ВcC, их гетерогенность не позволяют проектировщикам создавать систему «с нуля», требуя от них применения ранее накопленного опыта в виде успешных в прошлом технических решений. Эти же особенности заставляют, по мнению многих исследователей, повышать уровень абстракции, осуществляя проектирование путем правильной композиции уже готовых (или в достаточной мере описанных на данном уровне абстракции) компонент. По такому принципу, например, в настоящее время строится проектирование SoC: основная масса компонент системы представляет собой готовые блоки IP, которые объединяются в разработанную проектировщиками структуру. Сами IP при этом могут подвергаться незначительной «адаптации» под систему. Очевидно, применение паттернов как элементов накопленного успешного опыта должно осуществляться на всех уровнях абстракции проектируемой системы, в особенности там, где осуществляется основной объем проектирования – на самых верхних уровнях (системном, архитектурном [2]). Несмотря на это, практически все попытки разработки, спецификации и систематизации паттернов проектирования ВcC в настоящее время предполагают узкую специализацию под средства разработки (языки программирования, аппаратные платформы, элементную базу). Многими (в том числе и Дамашевичусом [3]), тем не менее, подчеркивается необходимость описания и применения паттернов высокоуровневого проектирования.

Программные и аппаратные решения

Встроенные системы в большинстве своем характеризуются тесным сопряжением *программных* и *аппаратных* решений. Проектирование ВcC практически всегда включает реализацию как программной, так и аппаратной части. Программное обеспечение встроенных систем взаимодействует с аппаратурой – датчиками и исполнительными устройствами. Встроенные системы создаются с целью взаимодействия с физическим миром – сбора информации и управления объектами посредством исполнительных устройств. Часть этого взаимодействия (отдельные его элементы) реализуется при проектировании аппаратным, а часть программным обеспечением. Это разделение функциональности обычно может иметь при проектировании аппаратуры несколько вариантов, так что выбор конкретного из них зависит от решений проектировщика и доступной ему элементной базы. Современные методы предлагают осуществлять основной объем проектирования на уровнях, не зависящих от программно-аппаратной дихотомии, проясняя во многих случаях вопрос разделения HW/SW и «отодвигая» это разделение на более поздние этапы проектирования, когда система почти готова в виде спецификаций.

Зачастую поведение компонента на системном уровне вообще не зависит от того, как он будет реализован – программно или аппаратно. Спецификацию его поведения можно отобразить как на «программную», так и на «аппаратную» платформу [2], причем с одинаково устраивающими проектировщика выходными характеристиками. Кроме того, в последнее время становится все сложнее отличить программное обеспечение от аппаратного, особенно с развитием технологий реализации аппаратных проектов на ПЛИС. Применение языков описания аппаратуры, таких как VHDL, Verilog (в особенности SystemVerilog), SystemC, JHDL, фактически стирает четкую грань между «программированием» поведения и созданием аппаратного блока из имеющихся в ПЛИС ресурсов. В случае применения средств высокоуровневого описания (например, при проектировании на Esterel или в САПР SCADe) спецификация компонента может быть автоматически синтезирована как в ПО, так и в аппаратуру. С точки зрения автора, при проектировании встроенных систем правильнее говорить не о «программности» или «аппаратности» компонента, а о степени последовательности либо параллельности реализуемых им вычислений.

Между тем большинство исследованных автором паттернов, относящихся к проектированию ВcC, четко формулируют свою принадлежность к «программному» или «аппаратному» контексту. Очевидно, такие паттерны найдут ограниченное применение в методиках проектирования, делающих акцент на уровне абстракции, инвариантные к HW/SW дихотомии.

Аспектная технология. Вычислительный механизм

В рамках аспектной технологии проектирования ВcC [2] предложен взгляд на проектируемую систему с разных точек зрения (в рамках различных *аспектов*). Каждый такой аспект учитывает лишь определенный ряд взаимосвязанных свойств системы. Примерами аспектов могут служить вычислительный или поведенческий, структурный, функциональный, конструктивный, надежностный, инструментальный, экономический аспекты. В каждом конкретном случае для проектировщиков может оказаться важным произвольный набор таких аспектов, причем суть любого из них (состав свойств системы в рамках аспекта) также может быть произвольной. Принципиальным является лишь «полнота» аспекта, т.е. самодостаточность получаемых в его рамках моделей системы. При таком подходе возможна независимая разработка системы целиком в рамках одного из ее аспектов, например, полная спецификация поведения системы в терминах некоторой вычислительной платформы, т.е. ее *реализация* на выбранной платформе.

Процесс проектирования ВcC в рамках этой технологии, таким образом, охватывает и рассматривает различные сферы деятельности коллектива проектировщиков: не только разработку непосредственно вычислительной ее части, но и применение определенных инструментальных технологий, «процессуальное» обеспечение проектирования (спецификация требований, контроль версий, учет дефектов и изменений), другие виды деятельности. Зачастую такие виды деятельности, остающиеся как бы на втором плане по отношению к разработке поведения системы, являются важными не менее самой разработки и во многом определяют ее успех. В рамках этих «сопряженных» с проектированием поведения видов деятельности у каждого коллектива, очевидно, есть некоторый накопленный опыт, в том числе и положительный – применяемые технологии внутрисхемного программирования, отладки аппаратных проектов, удачные конструкторские решения, «любимая» система контроля версий, стратегия развития и продвижения на рынок продукта. Паттерны как средство спецификации и сохранения положительного опыта могли бы выступать еще одним инструментом проектирования в рамках соответствующего аспекта. Применяя паттерны проектирования в рамках некоторого аспекта системы (например, удачный вариант размещения компонентов на печатной плате, который был применен ранее для платы с аналогичным составом компонентов), можно добиться эффекта, аналогичного эффекту от паттернов в области ООП. Грамотное применение собственных паттернов для различных аспектов одной и той же системы позволит значительно повысить эффективность проектирования.

Основное внимание в проектировании все же уделяется поведенческому аспекту ВcC. В рамках этого аспекта решаются проблемы организации вычислительного процесса, взаимодействия различных компонент системы в части обмена инструкциями и данными. Аспектная технология проектирования ВcC предлагает термин *вычислительный механизм*, который, хотя и употребляется в [2], но не имеет четкого определения обозначаемого им понятия. В [6] вычислительные механизмы понимаются как «технические решения из различных областей в абстрактном представлении». Далее автор статьи проясняет смысл сказанного: «...основываясь на приведенном выше определении механизма, данная абстракция играет роль основного «кирпичика» в организации вычислительного процесса. <...> вычислительный механизм решает частную задачу,

например, буферизация данных, декодирование управляющего слова, выборка и анализ команды, кэширование информационного потока». Вычислительный механизм, таким образом:

- решает некоторую частную техническую задачу;
- специфицирован в абстрактном виде, т.е. предназначен для отображения на конкретные случаи решения задач;
- является элементом организации вычислительного процесса в системе.

Исходя из вышесказанного, очевидна близость, если не идентичность, понятий *паттерна проектирования* и *вычислительного механизма* в контексте поведенческого аспекта ВСС. Предлагается рассматривать эти два термина в данном контексте как взаимозаменяемые.

Спецификация вычислительного механизма

Множество предлагаемых паттернов проектирования ВСС, в особенности попытки применения паттернов к проектированию аппаратуры, используют для спецификации задачи и ее решения нотацию UML, которая широко используется при спецификации паттернов в ООП. Очевидно, семантика языка заставляет разработчика подстраивать элементы HW проекта под понятия ООП: класс, объект, наследование, интерфейс и т.д. Во многих случаях наблюдаются попытки расширить нотацию UML там, где невозможно отождествить понятия аппаратного обеспечения и ООП. Существуют, однако, паттерны, в которых используются более адекватные языки [1]. Аспектная технология и некоторые другие методы проектирования [2] используют понятие *модели вычислений* для отражения системы правил, понятий и средств, в рамках которых организуется вычислительный процесс. В рамках модели вычислений для формальной спецификации системы выбирается наиболее адекватный язык, в терминах которого отображается вычислительный процесс (например, известные графические диаграммы для представления конечных автоматов Мили или Мура или текстовый язык Esterel для спецификации синхронно-реактивной системы). Очевидно, и спецификация вычислительного механизма должна содержать описание технического решения в терминах выбранной модели вычислений. Это, в свою очередь, позволит напрямую применять вычислительные механизмы в конкретных моделях, подвергая их формальное описание лишь незначительной адаптации под конкретную задачу. Язык, используемый для спецификации, при этом может быть любым подходящим для применения в рамках модели вычислений: преобразования описаний в другой синтаксис в этом случае не должны быть сложными и могут осуществляться как вручную проектировщиком, так и автоматически трансляторами (при условии формальности синтаксиса).

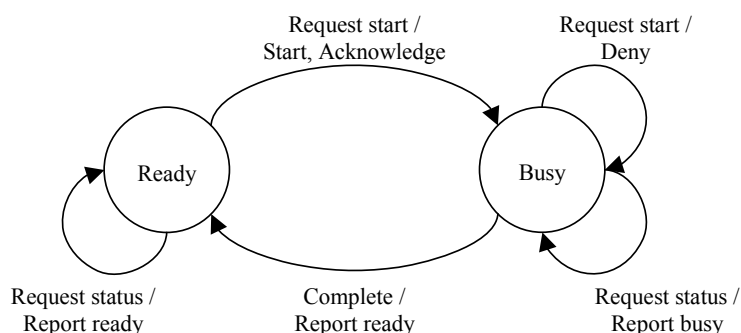


Рисунок. Асинхронный интерфейс компонента, выполняющего максимум одну операцию одновременно

На рисунке приведен пример спецификации вычислительного механизма, отражающего внешний интерфейс некоторого компонента системы, который способен выполнять максимум одну операцию.

Интерфейс позволяет компоненту выполнять операции асинхронно по отношению к использующим его компонентам-клиентам, а им, в свою очередь, не дожидаться завершения выполнения операции после ее запроса. Для спецификации механизма используется модель вычислений конечного автомата Мили, сам автомат представлен диаграммой переходов между состояниями.

В соответствии с механизмом компонент может находиться в двух состояниях: готов к выполнению операции («Ready») и выполняет операцию («Busy»). Переходы между этими состояниями осуществляются по следующим условиям.

- Request start – запрос начала операции от клиента.
- Request status – запрос состояния выполнения операции от клиента.
- Complete – завершение выполнения операции.

При выполнении переходов компонент производит следующие действия.

- Start – начать выполнение запрошенной операции.
- Acknowledge/Deny – подтвердить/отклонить запрос.
- Report ready/busy – сообщить о готовности либо неготовности к выполнению.

Очевидно, механизм не оговаривает особенности и количество выполняемых операций, способ передачи данных о запросе и о результате, состав этих данных, количество клиентов и возможность одновременного поступления запросов, а также многие другие особенности. Однако это и не требуется. Механизм описывает абстрактный принцип построения внешнего интерфейса компонента, конкретное наполнение этого интерфейса осуществляется в результате применения механизма при проектировании той или иной системы. Механизм в достаточной степени абстрактен, так, что он может описывать интерфейс как компонента, потенциально реализованного программно, так и потенциально аппаратного компонента. В самом деле, *Request start* и *Request status* могут быть программными вызовами некоторого модуля, результатами которых могут являться коды возврата вызовов соответственно *Acknowledge/Deny* и *Ready/Busy*. Действие *Report ready* в результате окончания выполнения операции может являться программным callback-вызовом клиента, запросившего выполнение операции. В другом же случае действия *Request start* и *Request status* могут быть соответственно записью определенного значения и чтением из порта периферийного блока микропроцессора. При этом компонентом, к внешнему интерфейсу которого применяется механизм, является периферийный блок, а клиентом – программный модуль-драйвер блока. Действие *Report ready* после завершения операции может быть запросом на прерывание к микропроцессору.

С другой стороны, механизм на рисунке в достаточной степени конкретен, чтобы зафиксировать отношения, которые могут возникнуть между компонентом и его клиентом. В частности, компонент не может выполнять одновременно более одной операции: после перехода в состояние «Busy» на любой новый запрос компонент выдает отказ. Также механизм предусматривает применение в тех случаях, когда не предполагается возникновение ошибки при начале или в ходе выполнения операции: на запрос о начале в состоянии «Ready» компонент всегда подтверждает начало операции, а после начала операция всегда завершается. Кроме того, механизм не предусматривает прерывание операции после начала ее выполнения. Таким образом, спецификация механизма на рисунке накладывает некоторые довольно конкретные ограничения на отношения компонента и его клиентов, фиксируя способы взаимодействия с ним в системе. Применение механизма, следовательно, представляет собой процесс фиксации определенных свойств вычислительного процесса, что, в свою очередь, является элементарным действием при проектировании поведенческого аспекта ВСС.

Заключение

Хотя аспектная технология проектирования ВвС находится в настоящее время на этапе своего формирования, она уже может быть признана по крайней мере перспективной [2, 6]. Многие вопросы, обсуждаемые в рамках данного метода проектирования, до сих пор не решены. В частности, нет четкого определения понятию «вычислительный механизм». В данной работе предлагается отождествить это понятие с паттерном проектирования в рамках поведенческого аспекта ВвС.

В работе дается обзор паттернов проектирования ВвС. В настоящее время применение паттернов в области встроенных систем довольно ограничено. Среди причин, ограничивающих применение и развитие этого довольно мощного инструмента проектирования, можно назвать, по крайней мере, две:

- четкое разделение задач (и, соответственно, решений) на «программные» и «аппаратные» на фоне развития методов проектирования ВвС, стирающих эту границу. Отсутствие спецификаций паттернов на системном, не зависящем от этого разделения, уровне, а потому, с развитием упомянутых методов, бесперспективность существующих паттернов;
- отсутствие способа формализации паттернов проектирования, обеспечивающего их применение в формальных спецификациях проектируемых систем. Существующие попытки адаптировать для этих нужд язык UML (объектно-ориентированный по природе) вынуждают разработчиков прибегать к различным ухищрениям. По мнению автора, для спецификации решений в области ВвС, пусть и абстрактных, достаточно достаточное количество более адекватных формальных языков.

Предлагается повысить уровень абстракции задач в рамках проектирования, решения которых могут быть описаны паттернами. Накопление положительного опыта в области высокоуровневого проектирования с помощью паттернов позволит существенно увеличить его эффективность. Это также создаст механизм для сохранения, формализации, обмена и улучшения опыта решения задач, традиционно возникающих в проектировании ВвС. Кроме того, предлагается использовать для формализации вычислительных механизмов (паттернов проектирования поведения ВвС) существующие формальные языки в рамках выбранной в конкретном случае модели вычислений.

Литература

1. Michael J. Pont, Royan H.L. Ong, Chinmay R. Parikh, Ridwan Kureemun, Chen Pang Wong, William Peasgood, Yuhua Li. A Selection of Patterns for Reliable Embedded Systems / EuroPlop '99, Kloster Irsee, 1999, <<http://www.le.ac.uk/engineering/mjp9/europlop99mjp.pdf>>
2. Постников Н.П. Поведенческий и инструментальный аспекты проектирования встроенных вычислительных систем; дисс. канд. технических наук – СПбГИТМО (ТУ), 2004.
3. Robertas Damaševičius, Giedrius Majauskas, Vytautas Štuikys. Application of Design Patterns for Hardware Design / 40th Design Automation Conference, Anaheim, 2003
4. E. Gamma, R. Helm, R. Johnson, J. Vlissides. Design Patterns: Elements of Reusable Object-Oriented Software. – Addison Wesley, 1995.
5. Jordi Cortadella, Mike Kishinevsky, Bill Grundmann. Synthesis of Synchronous Elastic Architectures / Proceedings of 43rd Design Automation Conference, ACM Press, 2006.
6. Платунов А.Е. Роль вычислительных моделей и механизмов в проектировании встроенных систем // Научно-технический вестник СПбГУ ИТМО. Выпуск 14. Информационные технологии, вычислительные и управляющие системы. СПб: СПбГУ ИТМО, 2004.

ОБНОВЛЕНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ В РАСПРЕДЕЛЕННЫХ УПРАВЛЯЮЩИХ СИСТЕМАХ

Е.В. Петров

Статья посвящена проблеме замены программного обеспечения элементов распределенных управляющих систем. В ней рассматриваются основные принципы построения загрузчиков таких систем, предлагается вариант их классификации по различным критериям. Рассматривается пример обновления программного обеспечения в системе управления городским освещением.

Задача обновления программного обеспечения возникает в любой системе, имеющей программную составляющую, независимо от того, является ли эта система распределенной, управляющей или встроенной, так как даже после сдачи готовой системы в эксплуатацию возникает необходимость замены программного обеспечения элементов системы. Это может происходить по нескольким причинам – исправление обнаруженных в программе ошибок, добавление новых функций в систему в связи с ее развитием и т. д.

Если элементов системы, нуждающихся в обновлении программного обеспечения (ПО), немного (несколько единиц), то замену ПО можно провести вручную. Данный способ требует наличия квалифицированных специалистов, способных выполнить действия по перепрограммированию контроллера узла системы и дополнительного оборудования: компьютера, источника питания и т.д., необходимого для этой операции. Если же система состоит из множества (десятков, сотен) элементов, появляется ряд дополнительных проблем: узлы системы могут находиться в различных районах города или даже за его пределами, а также доступ к узлам системы может быть затруднен. Поэтому применение ручного способа замены ПО становится неэффективным и потенциально ненадежным, в связи с увеличивающимся влиянием человеческого фактора, а также достаточно дорогостоящим и длительным, так как требует транспорта и времени для перемещения между узлами системы.

Следовательно, необходим более простой, надежный, дешевый, быстрый и максимально автоматизированный способ замены ПО элементов системы.

Обновление программного обеспечения в распределенных управляющих системах имеет свои особенности, отличающие ее от аналогичной задачи для систем на персональных компьютерах. При решении задачи обновления программного обеспечения распределенной системы следует учитывать следующие особенности.

- Большое количество элементов системы, требующих обновления программного обеспечения.
- Территориальная распределенность элементов системы (узлы системы могут находиться на большом расстоянии друг от друга).
- Разнородность элементов системы (узлы системы могут иметь различные наборы аппаратных средств и выполнять разные функции).
- Совместимость версий программного обеспечения элементов системы.
- Сохранение работоспособности элементов системы во время обновления программного обеспечения (элемент должен продолжать выполнение своих функций во время перепрограммирования).
- Надежность и безопасность процедуры замены программного обеспечения элементов системы.

Независимо от конкретного способа замены ПО в узлах системы ключевую роль в этом процессе играет так называемый загрузчик контроллера, и именно он во многом определяет набор возможных способов обновления ПО элементов системы. Загрузчик – это специальная программа, находящаяся в загрузочной области памяти контроллера и выполняющая следующие функции:

- запись данных в пользовательскую область памяти;
- передача управления целевой программе.

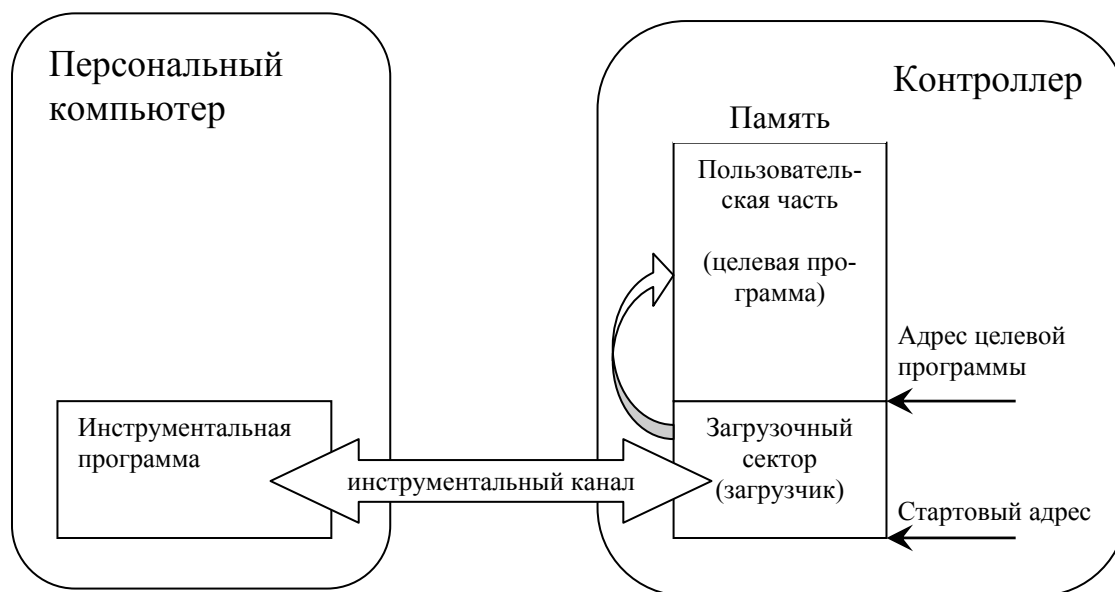


Рис. 1. Принцип работы загрузчика

На рис. 1 представлена типичная схема перепрограммирования контроллера. Программирование осуществляется с помощью персонального компьютера, подключенного к контроллеру через инструментальный канал. Инструментальным каналом может быть любой интерфейс, поддерживаемый и контроллером, и компьютером, например, RS-232, RS-485, USB, Ethernet и т.д., но наиболее часто в качестве инструментального канала используется последовательный канал (RS-232). После включения питания или перезапуска контроллера управление передается по стартовому адресу, по которому расположен загрузчик. Загрузчик, получив управление, пытается установить соединение с инструментальной программой, которая должна быть запущена на персональном компьютере. Назначение инструментальной программы – передача исполняемого кода через инструментальный канал в контроллер. Тип протокола взаимодействия инструментальной программы и загрузчика в каждом конкретном случае может быть разным (запрос–ответ, подтверждение в окне и т.д.). В том случае, если загрузчику удастся установить соединение с персональным компьютером, начинается процесс записи программного кода в пользовательскую память контроллера, в противном случае работа загрузчика прекращается. После окончания работы загрузчик передает управление по адресу целевой программы и начинается ее исполнение.

Из рис. 1 видно, что память контроллера разделена на две части – загрузочный сектор и пользовательскую область. Большинство микроконтроллеров поддерживает такое разделение на аппаратном уровне. В зависимости от типа микроконтроллера загрузочный сектор может располагаться либо в начале адресного пространства (в младших адресах), либо в конце (в старших адресах). Загрузочный сектор, как правило, имеет дополнительную защиту от случайного стирания, и именно в нем обычно располагается загрузчик. Таким образом, достигается дополнительная защита контроллера от некорректных действий пользователя. Даже после записи в микроконтроллер некорректной программы, благодаря независимому расположению загрузчика, сохраняется возможность записи в контроллер корректной программы без привлечения специальных программаторов.

Загрузчики можно разделить на несколько групп.

По назначению:

- Стартовый загрузчик, который записывает программу в исполняемую память (например, в оперативную) и передает ей управление. Кроме того, он может выполнять ряд подготовительных действий, необходимых для работы целевой программы (распаковка, декодирование и т.д.).
- Загрузчик программных модулей, который записывает исполняемый модуль в память по стартовому адресу и передает ему управление.
- Программатор памяти, который обеспечивает запись/чтение в память массива данных по указанному адресу. т.е. является гибким инструментом для доступа к внутренней памяти микроконтроллера и других модулей с программируемой памятью (EEPROM, DataFlash, FRAM и т.д.). Является удобным средством на этапе написания и отладки целевой программы.

По источникам данных:

- Данные загружаются из одного фиксированного локального канала передачи данных, как правило, RS-232 или RS-485.
- Данные загружаются из нескольких локальных каналов передачи данных, таких как RS-232, RS-485, I2C, SPI и т.д. Загрузчик прослушивает все доступные ему каналы передачи данных и реагирует на тот, по которому поступает сигнал начала программирования.
- Данные загружаются из нескольких различных каналов передачи данных, в том числе беспроводных (RS-232, RS-485, Ethernet, CAN, ZigBee, GSM/GPRS и т.д.).

По функциональным возможностям:

- Программный код загружается по фиксированному стартовому адресу в память.
- Блоки данных загружаются в различные программируемые устройства по указанным адресам.
- Загрузка производится с помощью специального загрузочного модуля, содержащего программируемые данные, а также описание последовательности действий, которые необходимо с ними произвести (в виде особой программы).

Технические характеристики загрузчика конкретного контроллера в большой степени зависят от имеющихся в данном контроллере ресурсов. Так, например, если микроконтроллер обладает всего несколькими единицами килобайт памяти программ, то реализовать в нем загрузчик, способный исполнять программы, наверное, не удастся. Скорее всего, это и не нужно: как правило, сложность контроллера и загрузчика связаны – чем сложнее контроллер, тем более сложный загрузчик в нем используется.

При построении системы обновления ПО необходимо сначала определить требования, которым она должна соответствовать, т.е. определить набор поддерживаемых функций и каналов передачи данных, с которыми будет взаимодействовать загрузчик, а затем соотнести эти требования с имеющимися аппаратными ресурсами (память, каналы передачи данных) и при обнаружении несоответствий внести изменения либо в требования, либо в аппаратуру.

В качестве примера систем обновления программного обеспечения рассмотрим систему загрузки обновлений ПО в СУНО (Система управления наружным освещением), разработанной и выпускаемой ООО «ЛМТ». Общая структура СУНО представлена на рис. 2. Система состоит из центрального диспетчерского пункта, в котором находится сервер, и множества электроподстанций, оборудованных контроллерами управления освещением. Узлы системы освещения связаны в структуру типа «звезда», центром которой является сервер СУНО. Узлами системы являются контроллеры в функции, которых входит установка заданного режима освещения, сбор и передача в ЦДП (центральный диспетчерский пункт) диагностической информации о состоянии подстанции и другие. Как правило, электроподстанции сети освещения распределены по террито-

рии всего города и находятся на больших расстояниях друг от друга, поэтому ручной способ обновления ПО контроллеров в данной системе является крайне неудобным.

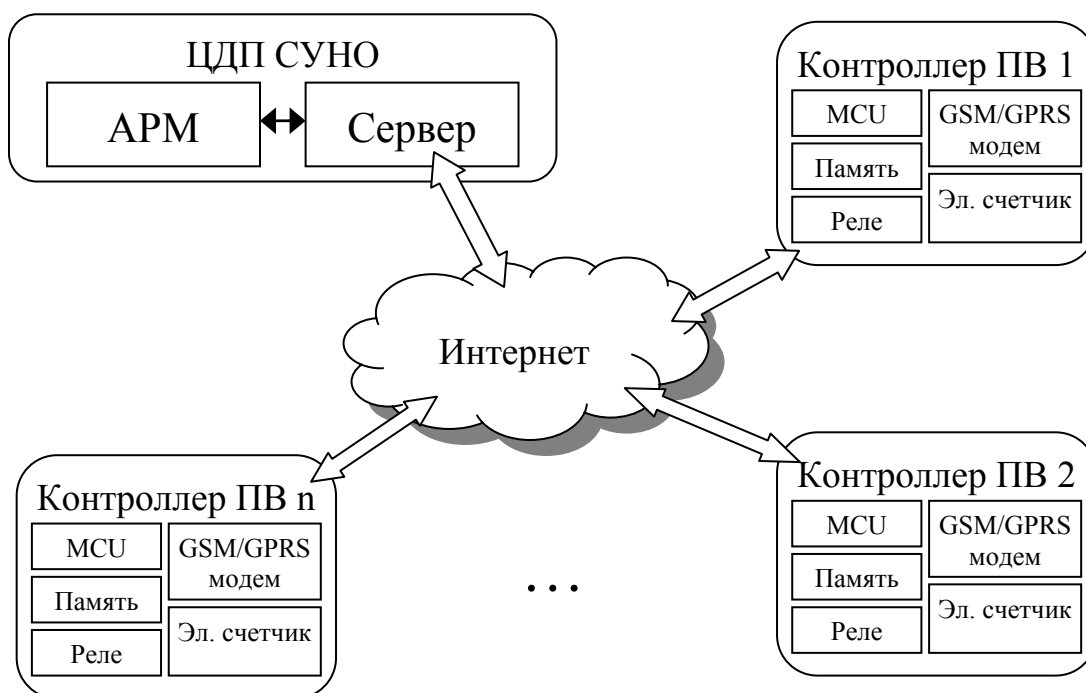


Рис. 2. Структура СУНО (ЦДП – центральный диспетчерский пункт, АРМ – автоматизированное рабочее место, ПВ – пункт включения)

К системе загрузки ПО СУНО выдвигались следующие требования:

- программирование контроллеров через последовательный канал;
- удаленное программирование контроллеров через GSM/GPRS;
- произвольный доступ к различным видам памяти контроллера (FLASH, FRAM);
- надежная замена ПО контроллера.

Аппаратура контроллеров позволяет реализовать все перечисленные требования: контроллер имеет инструментальный последовательный канал, GSM-модем, достаточные ресурсы памяти.

Система загрузки программного обеспечения СУНО состоит из двух больших частей – загрузчика контроллера пункта включения и серверной части. Загрузчик контроллера включает в себя все драйверы программируемых устройств, поддерживает обмен данными через последовательный канал и канал GSM, а также обеспечивает интерпретацию загрузочного модуля. Загрузочный модуль – это файл, состоящий из нескольких секций: сигнатуры, блока управления и блока данных. Сигнатура используется для идентификации загрузочного модуля в контроллере. Блок управления содержит программу, предписывающую, какие действия нужно выполнить с данными расположенными в блоке данных. Фрагмент такой программы на языке Forth приведен на рис. 3. Организация загрузчика как программируемой машины обусловлена тем, что такой способ обеспечивает максимальную гибкость и универсальность доступа к любым программируемым частям удаленного контроллера.

Набор команд загрузчика контроллера включает в себя около десяти инструкций, таких как ERASE – стереть указанный сектор памяти, COPY – скопировать из памяти-источника в память-приемник указанное количество байт, CHK – проверка корректности блока данных и др. Данная программа выполняет стирание семи секторов памяти программ, копирование из промежуточной памяти в основную память целевой программы, а также верификацию записанных данных.

REG_STRGT	TRGT_FLASH_P	LD
REG_DTRGT	TRGT_FLASH_P	LD
0x100		VERSION_CON
CHK		
SECTOR0		ERASE
SECTOR1		ERASE
SECTOR2		ERASE
SECTOR3		ERASE
SECTOR4		ERASE
SECTOR5		ERASE
SECTOR6		ERASE
0x20000000	spcm3_offset @ + SECTOR0	spcm3_len @ COPY
0x20000000	spcm3_offset @ + SECTOR0	spcm3_len @ COMPARE

Рис. 3. Пример программы для загрузчика контроллера ПБ

В функции серверной части входит подготовка загрузочного модуля и передача его в контроллер пункта включения. В понятие подготовки загрузочного модуля входит: трансляция программы с языка Forth в машинный код, понятный загрузчику; объединение всех программируемых данных в единый блок данных; объединение программы и блока данных в загрузочный модуль. Кроме того, для каждой секции загрузочного модуля подсчитывается и записывается CRC, что обеспечивает защиту контроллера от записи искаженных во время передачи данных. После того как загрузочный модуль подготовлен, он может быть передан в пункты включения. Так как все элементы системы, требующие замены ПО, имеют непосредственное соединение с сервером, это избавляет от необходимости использовать алгоритмы распространения ПО между пунктами включения. С другой стороны, это означает, что сервер должен следить за тем, чтобы все пункты включения получили новое ПО.

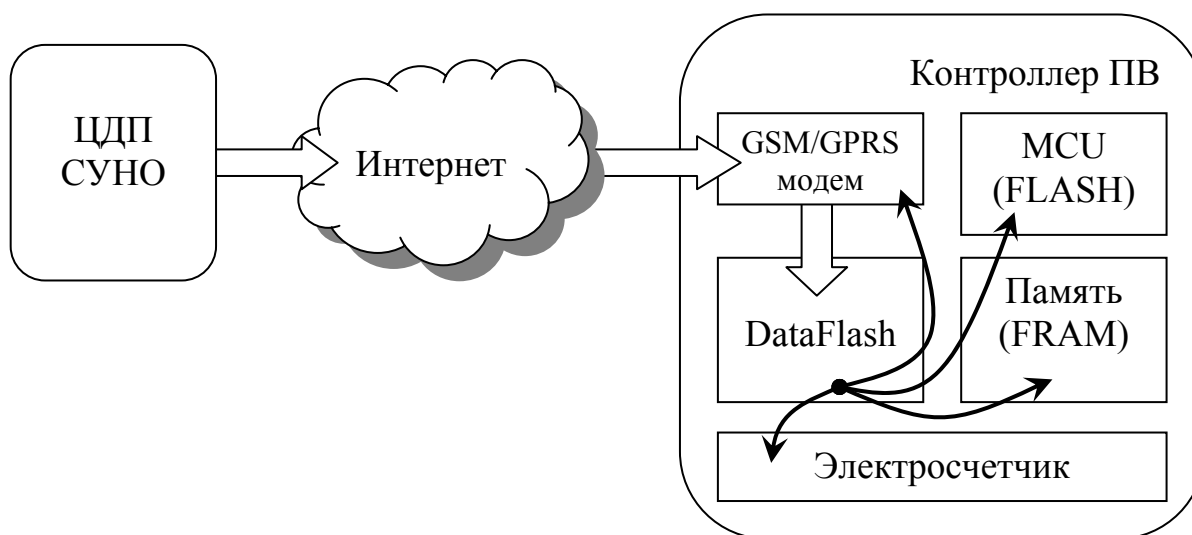


Рис. 4. Система обновления программного обеспечения СУНО

Схема обновления ПО на примере одного пункта включения показана на рис. 4. Замена ПО производится в три этапа:

- доставка загрузочного модуля в пункт включения;
- перезапуск пункта включения;
- исполнение загрузочного модуля контроллером пункта включения.

Прием загрузочного модуля на стороне пункта включения производится в специальную промежуточную память (DataFlash), поэтому, даже если по каким-либо причинам процесс доставки прервется, ПО пункта включения не будет повреждено. После того как загрузочный модуль был полностью принят и сохранен в DataFlash, производится перезапуск пункта включения.

Третий этап начинается с того, что загрузчик контроллера проверяет корректность сигнатуры и CRC секций загрузочного модуля. Если на каком-либо шаге проверки обнаруживается ошибка, исполнение загрузочного модуля прекращается, и управление передается целевой программе. Таким образом, обеспечивается высокая надежность и безопасность замены ПО контроллера. Если же ошибок не обнаружено, начинается исполнение программы, а после этого управление передается обновленной целевой программе.

Описанная выше система обновления программного обеспечения используется в нескольких городах России и позволяет экономить массу времени и средств при возникновении необходимости перепрограммирования удаленных контроллеров. Данная система загрузки может использоваться и как средство доставки данных в узлы системы. Для этого требуется только составить соответствующую программу для загрузчика, скомпилировать загрузочный модуль и передать его в соответствующий узел системы.

Перспективным развитием системы загрузки программного обеспечения систем подобных СУНО является включение в их функции возможностей по замене ПО не только самого контроллера, но и других устройств, подключенных к нему, например, модема, электросчетчика, сервисных процессоров и других.

Литература

1. Обновление системного программного обеспечения распределенных встроенных систем / Постников Н.П.
2. Network Reprogramming / Jaemin Jeong, Sukun Kim and Alan Broad // University of California, Berkeley. Aug 12, 2003
3. Downloadable Firmware in a Flash / Bill Gatliff // <URL: <http://www.embedded.com>>, Oct. 31, 2002

СИНХРОНИЗАЦИЯ ВРЕМЕНИ В РАСПРЕДЕЛЕННЫХ ИНФОРМАЦИОННО-УПРАВЛЯЮЩИХ СИСТЕМАХ

Д.Р. Ковязина

Применение распределенных информационно-управляющих систем (РИУС) в различных областях становится все более популярным. Используются современные высокоскоростные проводные и беспроводные технологии связи, электронные компоненты с высокой вычислительной мощностью, операционные системы реального времени и другие технологические приемы, однако в решении вопроса синхронизации времени в РИУС не поставлена точка. В статье обсуждаются основные проблемы обеспечения точного астрономического времени в РИУС, предлагаются способы их решения.

Введение

Время является одной из важнейших характеристик распределенной системы. Еще в конце 1970-х гг. Лесли Лэмпорт определил, что систему можно назвать распределенной, если время передачи сообщения является значительным по сравнению с интервалом времени между событиями в одном процессе [3]. В общем случае время является атрибутом события и может быть глобально или частично упорядоченным, приводя к соответствующей упорядоченности событий в системе. Можно сказать, что обеспечение правильной хронологии событий, происходящих в узлах распределенной системы, является первой задачей синхронизации времени, которой было уделено много внимания в начале пути создания таких систем. Второй задачей синхронизации времени является обеспечение точного астрономического времени во всей системе. Если первая проблема более характерна для распределенных универсальных систем с их транзакционными взаимодействиями между узлами, то вторая проблема – для распределенных информационно-управляющих систем (РИУС). В конечном счете, с вопросом синхронизации времени в РИУС тесно связан вопрос реального масштаба времени.

Примеры задач, решаемых РИУС, для которых синхронизация времени необходима: определение местоположения объекта, скорости его движения, слежение за параметрами окружающей среды (свет, температура, влажность, давление), идентификация узлов. С одной стороны, для выполнения такого рода задач к РИУС предъявляются более высокие требования по точности времени (иногда порядка микросекунд) по сравнению с распределенными универсальными системами, масштабируемости используемых методов синхронизации, надежности и безопасности, пониженного энергопотребления. С другой стороны, РИУС уступают распределенным универсальным системам в технологической оснащенности, выражающейся в применении достаточно скромных технологий программирования, инструментальных средств в сочетании с ограниченными ресурсами.

В данной статье обсуждаются проблемы обеспечения точного времени в РИУС и методы их разрешения.

Основные понятия

Синхронизация времени связана с пониманием временного порядка событий, генерируемых параллельными процессами, происходящими одновременно [4]. Это утверждение лежит в основе концепции логических часов. Сообщения, передаваемые между процессами, сопровождаются порядковыми номерами или временными метками, чтобы все взаимодействующие процессы могли «договориться» по поводу порядка следования событий.

Не всегда бывает достаточно временных меток, определяющих очередность событий [4]. В некоторых ситуациях нужно реальное время, поэтому появилась концепция физических часов. Основная проблема состоит в невозможности идентичной работы часовых генераторов в различных узлах распределенной системы. Корректность по-

казаний локальных часов определяется в результате сравнения их с источниками точного времени (опорные часы, эталонные часы). Под точным реальным временем обычно понимается всеобщее скоординированное время (UTC).

В узлах распределенной системы имеют место часовые ошибки двух видов: разница рабочих частот (*skew*) и разность во времени между локальными и опорными часами в любой момент времени (*offset*), что обусловлено разной частотой и сдвигом во времени установки часов. Таким образом, $skew = \frac{d(offset)}{dt}$. Под стабильностью (*stabil-*

ity) обычно подразумевается степень постоянства задающей частоты часов. При этом стабильность может быть двух видов: краткосрочная и долгосрочная. Краткосрочная нестабильность или дрейф (*drift*) чаще всего вызвана изменениями условий окружающей среды (температура, давление, влажность, механические изменения). Можно сказать, что $drift = \frac{d(skew)}{dt}$. Долгосрочная нестабильность связана с эффектом старения

кварцевого резонатора часов. Обычно единицей измерения дрейфа часов являются миллионные доли (*ppm – part-per-million*). Например, если дрейф часов равен ± 20 ppm, то это значит, что в месяц часы замедляются или убабстрыаются на $60 \frac{с}{мин} \cdot 60 \frac{мин}{ч} \cdot 24 \frac{ч}{день} \cdot 30 \frac{дней}{месяц} \cdot (\frac{20}{1000000}) = 51,84 \frac{с}{месяц}$. Под точностью локальных

часов (*ассигасу*) понимается степень соответствия их показаний времени опорных часов. Разрешение (*precision*) – минимальная единица времени, которую отсчитывают часы [6]. Синхронизация часов – синхронизация частоты и времени по эталонным часам.

В общем случае время $T(t)$, которое показывают часы в момент реального времени t , можно определить следующим образом:

$$T(t) = T(t_0) + R(t_0)[t - t_0] + \frac{1}{2}D(t_0)[t - t_0]^2, \quad (1)$$

где $T(t_0)$ – время, которое показывали часы в предыдущий момент времени t_0 , $R(t_0)$ – частота часов в момент времени t_0 , $D(t_0)$ – дрейф [1].

В корректировке физических часов есть три проблемы [4].

1. Выявление эталонных часов: атомные часы, GPS, радиочасы и т. д. Применение таких опорных часов для синхронизации времени обеспечивает очень высокую точность локальных часов. С другой стороны, непосредственное использование таких источников в распределенной системе неэкономично, а в некоторых случаях невозможно (GPS-приемники не могут быть использованы под водой, в закрытых помещениях). Поэтому для выполнения функции синхронизации времени были определены серверы времени в качестве опорных часов и алгоритмы корректировки физических часов в узлах распределенной системы по данным источникам (протокол синхронизации времени NTP/SNTP).
2. Если часы спешат или опаздывают, то установка в один момент правильного времени приведет к нарушению временного порядка следования событий в узле и системе в целом, так как это связано с немедленным переводом часов назад или вперед соответственно. Таким образом, следует использовать процессы замедления или убабстрения часов.
3. Ненулевое время прохождения сообщения с временной меткой при выполнении синхронизации времени.

Такие проблемы характерны и для распределенных универсальных систем, и для РИУС. Только первого вида системы столкнулись с этими проблемами раньше в силу своего возраста, и предложенные решения уже прошли этап апробации. Применение данных решений в РИУС требует их изменения в связи с более высокими требованиями точности, надежности, безопасности, гибкости. Кроме того, в РИУС отсутствует из-

быточность опорных источников и связей между узлами, что характерно для распределенных универсальных систем.

Длительность передачи сообщения

Все методы синхронизации должны учитывать, что время прохождения сообщения содержит четыре базовых компонента [2]: время передачи (send time), время доступа (access time), время распространения (propagation time), время приема (receive time). Время передачи – время создания сообщения для отправки в сеть передатчиком. Чаще всего оно включает в себя временные задержки, источником которых является работа операционной системы (смена контекста, обработчики прерывания, системные вызовы и др.), и время, затрачиваемое на прохождение сообщения с прикладного уровня на сетевой уровень. Время доступа – время ожидания доступа в сеть (среда передачи), задержка на сетевом уровне. Например, в случае Ethernet это ожидание освобождения канала, в случае использования коммутации каналов на основе разделения времени (TDMA) это ожидание сообщением своего тайм-слота. Время распространения – время, затрачиваемое на физическую передачу битов сообщения в среде от передатчика до приемника. Время приема – время обработки полученного сообщения приемником.

Таким образом, главная проблема синхронизации состоит не в том, что эти временные задержки существуют, а в том, что их очень сложно предсказать, просчитать для каждой отправки. Исключение любой из этих компонент заметно увеличит производительность метода синхронизации. В глобальных сетях наибольшей составляющей длительности передачи сообщения является время распространения (задержки буферизации сообщений в маршрутизаторах), а в локальных сетях – время передачи и доступа. Поэтому для локальных сетей в недавнем прошлом были разработаны методы синхронизации, исключаяющие одну или обе эти составляющие, что позволило увеличить точность синхронизации времени. Это методы синхронизации времени RBS (Reference-Broadcast Synchronization), TPSN (Timing-sync Protocol for Sensor Networks), FTSP (Flooding Time Synchronization Protocol) для сенсорных сетей.

Протокол синхронизации времени NTP

Одним из самых популярных протоколов синхронизации времени в распределенных универсальных системах является Network Time Protocol (NTP). Назначение протокола состоит в синхронизации клиента или сервера с сервером или источником точного времени (радиочасы, атомные часы, GPS и др.). Синхронизируется не только текущее значение времени, но и частота отсчета таймера. Обеспечивается точность до миллисекунды в пределах локальной сети и десятков миллисекунд в пределах глобальной сети. Предусмотрена криптографическая защита, одновременное подключение к нескольким серверам на случай аварии, алгоритмы усреднения и т.д. Главные серверы (напрямую присоединенные к опорному источнику) образуют первый слой, присоединенные непосредственно к ним – второй слой, и т.д. Для обмена информацией между клиентом и сервером используется протокол UDP (порт 123). Применяются довольно сложные алгоритмы фильтрации, селекции и комбинации пакетов на принципах максимальной вероятности. Протокол обеспечивает поддержку множества резервных серверов и путей передачи (выбор лучшего на основе алгоритма взвешенного голосования). Типичный интервал опроса – от 1 минуты (в начале работы) до 17 минут (если все хорошо) [1]. Сервер непрерывно корректирует ход локальных часов, используя вычисленную информацию об отклонениях их частоты от истинной. Это позволяет уменьшить частоту опроса и удерживать отклонения показаний часов от истинных при временных сбоях сети. Подстройка частоты обеспечивает удовлетворительную точность

часов даже при модемном соединении с Интернет. При больших отклонениях местного времени от времени выбранного сервера коррекция производится скачком, иначе – путем подстройки частоты местных часов.

Протокол NTP предоставляет различные классы обслуживания, которые позволяют определить, кто (сервер или клиент) инициирует процедуру синхронизации и, собственно, как эта процедура организована. Возможны следующие классы обслуживания.

- Multicast: для использования в быстрой локальной сети с множеством клиентов и без необходимости в высокой точности один или более NTP-серверов рассылают широковещательные сообщения, клиенты определяют время, исходя из предположения, что задержка составляет несколько миллисекунд; сервер не принимает ответных NTP-сообщений.
- Procedure-call: в условиях, когда нужна высокая точность, а Multicast недоступен, NTP-клиент посылает NTP-запрос на сервер, который обрабатывает его и немедленно посылает ответ (рис. 1).

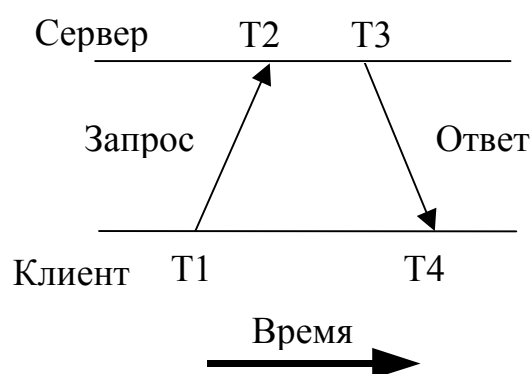


Рис. 1. Класс обслуживания Procedure-call

Протокол NTP создан с целью определения трех величин: временного сдвига (offset, θ), периода кругового обращения сообщения (Roundtrip delay, δ) и дисперсии (dispersion, ε). Все они вычисляются по отношению к выбранным эталонным часам. Смещение часов определяет поправку, которую необходимо внести в показания местных часов, чтобы результат совпал с показанием эталонных часов. Дисперсия характеризует максимальную ошибку локальных часов по отношению к эталонным [6]. В протоколе NTP дрейфом часов пренебрегают. Временной сдвиг и период кругового обращения сообщения рассчитывается следующим образом:

$$\theta = \frac{(T_2 - T_1) + (T_3 - T_4)}{2}, \quad \delta = (T_2 - T_1) - (T_3 - T_4) \quad (2)$$

где T_1 – время передачи сообщения клиентом, T_2 – время приема сообщения сервером, T_3 – время передачи сообщения сервером, T_4 – время приема сообщения клиентом.

Доверительный интервал настоящей величины θ_0 составляет

$$\theta - \left(\frac{\delta}{2} + \varepsilon\right) \leq \theta_0 \leq \theta + \left(\frac{\delta}{2} + \varepsilon\right), \quad (3)$$

где θ – рассчитываемая величина временного сдвига.

Таким образом, клиент получает сообщение с временными метками T_1 , T_2 , T_3 и сам делает временную засечку T_4 , рассчитывает δ , θ согласно формулам (2) и по ним корректирует свои часы. Данная концепция построена на допущении, что время передачи сообщения-запроса равно времени передачи сообщения-ответа. Для повышения точности данного протокола выполняют временные отметки на сетевом уровне, а не на прикладном.

Интересно концепцию NTP применить в РИУС, при этом не следует пренебрегать дрейфом часов. В следующем разделе обсуждаются методы решения описанных ранее проблем синхронизации времени в конкретной РИУС.

Пример распределенной информационно-управляющей системы

Примером системы, в которой задача синхронизации времени является важной, может послужить система управления наружным освещением (СУНО). СУНО предназначена для удаленного централизованного управления электротехнической аппаратурой наружного освещения населенных пунктов и для сбора диагностической информации о текущем режиме работы и состоянии аппаратуры уличного освещения. Система включает в себя центральный диспетчерский пункт (ЦДП), состоящий из сервера сбора данных и автоматизированного рабочего места (АРМ) оператора; пункты включения (ПВ), содержащие контроллер ПВ с модемом и электросчетчик. Канал связи между ЦДП и ПВ – это GPRS/GSM. СУНО поддерживает работу в автоматическом режиме или по командам оператора.

Задачи, для которых необходимо точное астрономическое время, – это автоматическое управление («Таймер») контроллера ПВ, ведение журнала событий, работа электросчетчика, подсоединенного к контроллеру ПВ. В режиме автоматического управления включение и отключение светильников наружного освещения осуществляется по годовому графику от встроенного таймера с часами реального времени. Именно от правильности времени включения и выключения освещения по графику зависит безопасность людей, расход электроэнергии. Годовой график включения и отключения светильников наружного освещения хранится в энергонезависимой памяти контроллера ПВ, его корректность определяется контрольной суммой. В журнале событий должна присутствовать информация о действиях оператора, работающего в ЦДП, телеметрия, полученная из ПВ, информация об авариях, произошедших на ПВ, и, конечно, к каждому этому событию дата и время. От точности настройки часов в электросчетчике зависит производимый им энергоучет в соответствии с тарифным планом.

Чтобы обеспечить точное время для решения всех этих задач, нужно сначала определить максимально допустимую величину временного сдвига (offset) для каждой задачи. Годовой график освещения может задаваться с минимальной дискретностью величиной в минуту. Тарифный план для электросчетчиков тоже задается с минутной точностью. События в журнале событий должны фиксироваться с точностью до 1 секунды. Таким образом, необходимо обеспечить секундную точность. Возможно ли это?

Источниками реального времени в системе можно назвать сервер сбора данных и АРМ, часы реального времени в контроллере ПВ, модеме ПВ и электросчетчике. Из них опорными источниками являются сервер сбора данных и АРМ при наличии связи с ПВ. В АРМе реализуется функция синхронизации времени в ПВ, при этом источником точного времени могут быть системные часы АРМа или сервера сбора данных (второй вариант предпочтительнее по точности), в свою очередь, данные системные часы могут быть синхронизированы по протоколу NTP с серверами времени.

Задача обеспечения надежной работы часов в ПВ связана, в первую очередь, с определением правильности хода часов реального времени в контроллере ПВ. Можно предложить следующие решения данной задачи.

1. Интервал времени (минута или несколько минут) замеряется на секундном (или более точном) таймере процессора ПВ и сравнивается с интервалом времени, отсчитанным за эту минуту (несколько минут) часами реального времени. Такое сравнение имеет смысл, если кварцевый резонатор процессора точнее кварцевого резонатора часов реального времени. Таким образом, можно определить дрейф часов

(сек./мин.), и если он превышает заданное значение, то показания часов признаются некорректными, и их нужно синхронизировать.

2. Должно производиться периодическое сравнение показаний часов реального времени контроллера ПВ, модема и электросчетчика при условии, что последние два источника точного времени тоже корректируются в течение процедуры синхронизации времени с ЦДП. При этом модем должен работать в мультиплексированном режиме, т.е. воспринимать и передаваемые по каналу GPRS/GSM данные, и команды. За точное время могут приниматься показания часов, выбранных по мажоритарному принципу: совпадение времени с определенным допуском у двух из трех часов. Если присутствует сильное расхождение показаний всех трех источников, то выполняется процедура синхронизации времени с ЦДП.
3. Определяется системная переменная, хранящая текущее время в секундах, начиная с базового времени (например, 1 января 2000 года 00:00:00). Данная переменная инициализируется при запуске системы по часам контроллера ПВ при условии, что часы на тот момент скорректированы. Каждую минуту (или другой интервал времени) показания часов реального времени, переведенные в секунды, сравниваются с этой переменной, в которой хранятся показания часов предыдущей проверки. Если разница между этими значениями не является положительным числом, то произошел сбой часов, и они требуют синхронизации времени.

Наличие нескольких источников точного времени в ПВ является необходимым условием обеспечения надежной работы ПВ. Топология СУНО является централизованной («звезда»), поэтому в случае отсутствия связи между ЦДП и ПВ предложенное ранее решение № 2 становится самым подходящим.

Как ранее было определено, задача синхронизации часов распадается на две подзадачи: синхронизация частоты и синхронизация времени (формула (1)). Задачу синхронизации частоты тоже можно разделить на два уровня. Первый уровень – корректировка стабильности частоты с использованием калибровочных коэффициентов с целью минимизации влияния факторов окружающей среды, главным образом, температуры, на работу часов. Вторым уровнем – уменьшение отклонения частоты локальных часов от опорных (skew). Последнее является достаточно сложной задачей из-за трудности измерения частоты часов в рабочем режиме.

Проблема резкости установки времени в часах является тоже актуальной для данной системы. Такая проблема вызывает нарушение хронологии событий, фиксируемых в журнале событий, и ошибочный учет электроэнергии электросчетчиком. Как говорится, время никогда не должно идти назад. В распределенных универсальных системах данную проблему решают при помощи замедления или убыстрения хода системных часов путем увеличения или уменьшения интервала системных вызовов прерываний таймера (например, если для работы системных часов прерывания вызываются каждые 17 мс, а часы отстают от точного времени, то 17 мс сменяются на 15 мс). Такую возможность предоставляет протокол NTP. В нашем же случае для замедления или убыстрения можно использовать калибровочные коэффициенты. Этот вариант тоже не всегда годится, так как данные коэффициенты позволяют откорректировать дрейф часов лишь в некотором диапазоне (например, $\pm 130 \text{ ppm} = \pm 5,6 \text{ мин./месяц}$, что составляет примерно $\pm 10 \text{ с/день}$). При этом дрейф часов можно определить при помощи способа № 1, описанного ранее и проверяющего правильность хода часов. Другим решением этой проблемы является реализация системных часов как в персональном компьютере на основе системного секундного таймера, описанного ранее в способе № 3, по проверке правильности хода часов. Как и в протоколе NTP, можно влиять на эту переменную, т.е. замедлять или убыстрять ее инкремент. Таким образом, источником точного времени в контроллере ПВ становится этот системный таймер, и все алгоритмы синхронизации нужно применять именно к нему в первую очередь. Конечно же, периодическое

сравнение показаний системных часов и трех остальных источников точного времени в ПВ должно осуществляться. Если расхождение во времени настолько велико, что предложенные способы подстройки частоты часов не решают проблемы, то следует просто установить точное время в часах, полученное с использованием алгоритма синхронизации.

Неопределенность длительности передачи сообщения с временной меткой является заметной проблемой. По изложенной выше модели длительности передачи сообщения между двумя узлами в данном случае можно сказать о том, что наибольшей составляющей этой задержки является время распространения сообщения (propagation time), а потом уже время доступа, время передачи и время приема. Время приема – время, затрачиваемое на обработку полученного сообщения модемом ПВ и вычитывание его из модема через последовательный канал на скорости 57600 бит/с процессором контроллера ПВ. В пункте включения нет операционной системы реального времени, не поддерживан стек TCP/IP, поэтому время приема является малой величиной, как и время передачи, время доступа по сравнению со временем распространения. Время распространения – время, отсчитываемое от момента покидания сообщением сетевой карты компьютера (или внешнего модема при использовании GSM-канала) ЦДП, через маршрутизаторы, рабочие и базовые станции сотовой связи на скорости 38400 бит/с, до приема модемом ПВ. Данные рассуждения предполагают наличие регистрации в GPRS-сети модема ПВ на момент начала процедуры синхронизации времени. Если это не выполнено, то данная процедура станет еще дольше на десятки секунд. Задержка передачи сообщения становится еще большей в случае большой загрузки базовых станций сотовой связи, когда сообщение ожидает освобождения тайм-слота для своей дальнейшей передачи.

При использовании резервного GSM-канала длительность передачи сообщения значительно увеличивается, так как сначала требуется дозвониться до ПВ ЦДП, а потом на скорости 9600 бит/с передать сообщение. Такой вариант получается очень дорогим (повременная оплата каждого соединения) в случае реализации автоматической синхронизации с некоторой периодичностью во времени, инициируемой ЦДП, при которой каждая процедура синхронизации должна состоять из обмена совокупностью сообщений с временными метками. Количество таких сообщений определяется требуемой точностью синхронизации часов (максимально допустимая величина временного сдвига). Неопределенность величины каждого компонента, составляющего длительность передачи сообщения, является основной проблемой. Можно сказать, что для более точной оценки времени распространения сообщения с использованием GPRS/GSM связи требуется проведение дальнейших исследований.

Для обеспечения минутной точности часов ПВ достаточно простой посылки сообщения с временной меткой из ЦДП в ПВ. Однако требование секундной точности обязывает учитывать длительность передачи сообщения в алгоритмах синхронизации времени. В таком случае можно использовать алгоритмы синхронизации, заложенные в классах обслуживания протокола NTP.

1. Как в классе обслуживания Multicast протокола NTP, часы ПВ корректируются по двум временным меткам: T_1 – время передачи сообщения ЦДП, T_2 – время приема сообщения ПВ. Посылка сообщений ЦДП производится до тех пор, пока не будет достигнута заданная точность часов ПВ.
2. Как в классе обслуживания Procedure-call протокола NTP, выполняется двунаправленный обмен сообщениями, и корректировка часов производится по четырем временным меткам (T_1, T_2, T_3, T_4) и рассчитанным по формулам (2) значениям временного сдвига, периода кругового обращения сообщения, дисперсии. Кроме того, следует учитывать доверительный интервал полученного временного сдвига (формула (3)).

Еще большего эффекта в смысле повышения точности корректировки часов можно добиться от описанных алгоритмов синхронизации, если сделать процедуру синхронизации автоматически повторяющейся с некоторым периодом. Так как между моментами установки точного времени часы ПВ рассинхронизируются под действием описанных ранее факторов, то для определения величины периода синхронизации часов следует оценить порядок рассинхронизации. К тому же следует учитывать трафик, вызываемый автоматической синхронизацией, и его стоимость. Поэтому менее затратным является вызов процедуры синхронизации по требованию ПВ при обнаружении большой неточности часов в результате проверки правильности их хода.

Из всех изложенных фактов следует, что требование минутной точности работы часов контроллера ПВ можно выполнить, а секундной – проблематично и требует более детальных дальнейших исследований.

Заключение

Проблема синхронизации времени в распределенных информационно-управляющих системах не имеет общепринятого решения в силу разнообразия и специфичности требований таких систем. По сей день предлагаются методы синхронизации времени для РИУС определенного вида, области применения (например, RBS, TPSN для сенсорных сетей). В статье уделено внимание одной из задач синхронизации времени – обеспечение точного астрономического времени; подробно рассмотрены основные трудности, сопровождающие решение этой задачи – выявление эталонных часов, резкость установки времени и ненулевое время передачи сообщения с временной меткой. На примере конкретной РИУС были продемонстрированы все эти проблемы и другие трудности, связанные с синхронизацией времени, и предложены методы их решения. Некоторые решения являются упрощенными вариантами методов синхронизации времени в распределенных универсальных системах, а некоторые специализированы под данную систему. Представленные способы обеспечения точного времени могут быть использованы в других РИУС сходной организации.

Литература

1. Mills D. L. Improved Algorithms for Synchronizing Computer Network Clocks. 1994.
2. Elson J., Girod L., Estrin D. Fine-Grained Network Time Synchronization using Reference Broadcasts. 2002.
3. Lamport L. Time, Clocks and the Ordering of Events in Distributed Systems. 1978.
4. Krzyzanowski P. Clock Synchronization. Lectures on distributed systems. 2000–2002.
5. Srikanth T. K., Toueg S. Optimal Clock Synchronization. 1987.
6. Семенов Ю. А. Сетевой протокол времени NTP. 2004.

РАЗРАБОТКА ИНФОРМАЦИОННО-УПРАВЛЯЮЩИХ СИСТЕМ КАК АКТУАЛИЗАЦИЯ ЦЕЛЕВОЙ ЗАДАЧИ

Р.Р. Ковязин

Потребность в проектировании информационно-управляющих систем (ИУС) продолжает активно расти. При этом процессы проектирования по-прежнему требуют совершенствования. Методики разработки программной и аппаратной компонент ИУС развиваются в разных направлениях, что не позволяет устранить противоречия между рабочими группами и вредит конечному результату – созданию системы. В статье предлагаются механизм и средства описания ИУС, объединяющие в себе фазы Design-Time и Run-Time и оперирующие составными частями ИУС инвариантно к их реализации.

Введение

Информационно-управляющие системы (ИУС) являются важнейшим элементом любой современной системы автоматизации. Область их применения распространяется на медицинскую и бытовую технику, телекоммуникации, транспортные и аэрокосмические системы и др. Такие отрасли, как энергетика, металлургия, теплоснабжение, не могут обойтись без средств автоматизации. Существующие системы автоматизации основываются на аналоговой и цифровой технике, релейно-контактных схемах, механических агрегатах. На текущий момент интенсивно развивающийся вид ИУС – это системы с программным управлением. Многие управляющие системы и системы автоматизации, организованные по другим принципам, модернизируются в системы с программным управлением. Примерами таких систем являются системы промышленной и железнодорожной автоматизации, системы энергоснабжения, измерительные приборы и др.

За последнее десятилетие многократно увеличилась вычислительная мощность электронных компонент, используемых в вычислительных системах, в том числе в ИУС. Это позволяет современным ИУС иметь сложную организацию, использовать операционные системы реального времени, объединяться в контроллерные сети (Embedded Net), применять современные высокоскоростные проводные и беспроводные технологии связи. Но все это не отменяет трудностей реализации характерных особенностей ИУС, потому что появление новых электронных компонент позволяет ставить перед ИУС новые, более сложные задачи. Повышение вычислительной мощности устройств реализации программного управления привело к тому, что на рынке стали востребованы ИУС со сложной функциональностью.

Вместе с прогрессом электронных компонент появились новые методики разработки компонент ИУС. То, что раньше было неприемлемо из-за сложности или неэффективности реализации, сейчас уже общепринято и широко распространено. Увеличение сложности ИУС привело к тому, что повысилась требуемая квалификация разработчиков компонент ИУС. Следствием этого стала более узкая специализация разработчиков, что разделило их на два «лагеря» и стало мешать эффективному созданию ИУС. Методики проектирования каждой из компонент ИУС ограничены и содержат мало или вообще не содержат рекомендаций по организации их совместной работы для реализации программного управления системы. Критерии качества компонент не соответствуют критериям качества системы в целом. В чем-то методики проектирования компонент даже противоречат друг другу, так что успешное проектирование системы есть компромисс между ее компонентами. К этому следует добавить, что методики разработки ИУС не поспевают за темпами роста вычислительной мощности электронных компонент и требованиями рынка.

Некоторые исследователи проблем проектирования ИУС, несмотря на очевидный прогресс в области их разработки, отмечают, что этот прогресс неправильный, это тупиковый путь [1, 2]. В данной работе представлены основные положения модели актуализации. Целью ее создания было получение описания ИУС, объединяющего в себе

фазы Design-Time и Run-Time, программную и аппаратную компоненты. Модель ориентирована на ИУС с доминирующей программной компонентой (Software-Based Embedded Systems).

Процесс актуализации

Первый документ, определяющий целевую задачу любой ИУС – это техническое задание (ТЗ). Для ИУС постановщиком ТЗ является прикладной специалист. ТЗ на разработку вычислительных систем всегда является неполным, поэтому в их разработке присутствует этап проектирования. ТЗ содержит информацию о том, «что» должна выполнять система (часто неформально) и «как» она должна это делать. Неполнота ТЗ заключается в том, что даются не все указания «как», а указания «что» допускают двоякую трактовку и реализацию. Потенциально разработчики свободны в способе реализации ИУС, лишь бы она выполняла требуемые от нее функции. Указания «как» являются ограничением этой свободы: элементная база, минимальная производительность, инструментальные средства, энергопотребление, стоимость, надежность и др. Задача этапа проектирования состоит в добавлении недостающих указаний «как» и в выборе конкретных «что».

ТЗ, дополненное всеми недостающими указаниями, назовем *целевым алгоритмом* (ЦА). Дополнить ТЗ можно по-разному, поэтому ЦА, реализующих ТЗ, может быть несколько. Задача разработчиков ИУС заключается в реализации конкретного ЦА вычислительными средствами ИУС. В научной фантастике человек может общаться с вычислительной машиной. К сожалению, вычислительные машины не понимают человеческого языка, хотя ведутся исследования в этом направлении. Пока что для реализации ЦА требуется применение промежуточных языков. В процессе передачи «желания» человека машине ЦА претерпевает ряд преобразований из одного языка в другой, часть из которых выполняют разработчики ИУС, а часть выполняется автоматизированно инструментальными средствами и самой ИУС. Очевидно, что этот процесс не является полностью автоматизированным, поскольку инструментальные машины, с помощью которых выполняются преобразования ЦА, тоже не понимают человеческого языка. Реализация ЦА заключается в выполнении функций ТЗ с учетом его ограничений. Весь процесс преобразований «запускается» не для создания ИУС, не для создания программы, а для того, что хочет получить человек. ЦА претерпевает множество изменений, пока не преобразуется в требуемые физические сигналы: свет, движение, электричество и др. Назовем процесс преобразования ЦА в «желаемое» человеком *процессом актуализации*.

В качестве модели процесса актуализации ЦА будем использовать ориентированный граф (*граф актуализации*). Процесс состоит из разветвленной системы преобразований информации, начинающейся с ЦА и заканчивающейся конечными физическими сигналами. Вершинами графа являются *трансляторы*, преобразующие информацию из одной формы представления в другую. Дуги графа – это формы представления ЦА. Таким образом, последовательность трансляторов, соединенных дугами, задает последовательность преобразований ЦА. Поскольку действия разработчиков ИУС не являются формальными, мы их исключим из графа актуализации. Исходным представлением информации будем считать *целевую программу* (ЦП) – ЦА, записанный на каком-нибудь языке (программирования), для которого есть инструментальные средства. Сувив граф актуализации, мы оставили в нем лишь автоматические трансляторы, что позволит рассмотреть его формально и получить численные характеристики. На рис. 1 представлен пример типичного графа актуализации для ИУС с программным управлением.

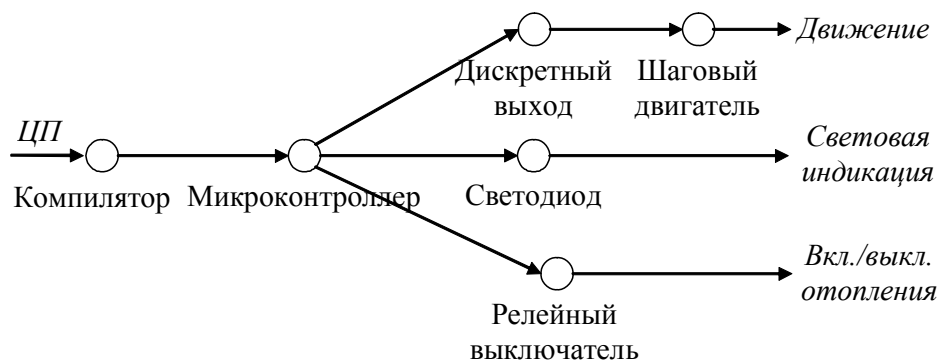


Рис. 1. Пример графа актуализации

Граф актуализации можно использовать для решения двух задач.

- Создание графа актуализации и ИУС по нему. Это является одним из способов проектирования ИУС.
- Создание графа по существующей ИУС. На основе графа можно получить характеристики для повторного использования трансляторов, модернизации и рефакторинга ИУС.

Предлагаемая модель не является одной из моделей жизненного цикла вычислительных систем [3]. Они моделируют этапы существования систем, а процесс актуализации соответствует всему процессу преобразования информации и является статическим. Трансляторы, участвующие в этом процессе, могут показаться принципиально разными, но это обусловлено несовершенством вычислительных возможностей современной элементной базы.

Трансляторы и потоки

Понятие *транслятор* базируется на TSM-модели [4]. У транслятора есть входные и выходные потоки: по входным он получает команды (сигналы), а по выходным генерирует реакции на команды. Транслятор будем называть *программируемым*, если он поддерживает входные сигналы, приводящие к появлению выходных сигналов (реакций). Транслятор будем называть *конфигурируемым*, если он поддерживает входные сигналы, приводящие к смене реакций на программирующие команды. Потоки, по которым передаются команды, приводящие к реакциям или к сменам реакций, будем называть *программирующими (программами)* и *конфигурирующими (конфигурациями)*. Любой поток состоит из команд одного типа. На графе актуализации программы будем обозначать сплошными дугами, а конфигурации – пунктирными. Трансляторы будем обозначать кругами. Для актуализации ЦП ИУС некоторым трансляторам даются программирующие команды, чтобы они выполнили действия. Некоторые из этих трансляторов не способны самостоятельно актуализировать полученные команды. Они разбивают команды на части и передают их для актуализации другим трансляторам. Так команды передаются от транслятора к транслятору до тех пор, пока не дойдут до тех трансляторов, которые могут сами актуализировать полученные команды. Их суммарные реакции как раз и составляют действия ИУС, которые она должна выполнять в соответствии с ЦП. Потоки являются формами представления частей ЦП, а трансляторы – объектами, преобразующими эти формы представления, разделяющими и соединяющими их части. Трансляторы преобразуют программы и конфигурации в новые программы и конфигурации.

Если один транслятор программируется или конфигурируется другим транслятором, то он является его *ресурсом*. Отношение между трансляторами можно определять не через необходимость актуализации их команд. Допустим, есть транслятор А с ресурсом Б, и

нужно получить реакцию ресурса. Непосредственно программировать ресурс Б невозможно, потому что к нему нет доступа. Но его можно программировать через транслятор А. Даже если известны языки входных потоков ресурса Б, ему можно давать не любые команды, а только те, которые для него генерирует транслятор А, обрабатывая собственные входные команды. Аналогичным образом можно передавать команды через цепочку трансляторов. Все трансляторы, к которым можно получить доступ через команды транслятора, являются его непосредственными или опосредованными ресурсами.

Часто ИУС получают информацию из своего окружения – температура воздуха, напряжение во входных электрических цепях, действия оператора и др. С одной стороны, эти сигналы можно разделить на программирующие и конфигурирующие. С другой стороны, потоки, по которым передаются эти сигналы, не являются формой представления ЦП. Получается, что у некоторых трансляторов не все входные потоки являются формой представления ЦП. Назовем такие трансляторы *смешанными*, а такие потоки *внешними*. Транслятор, у которого присутствуют только внешние потоки, назовем *0-транслятором*. Не следует путать, например, кнопку или датчик с 0-транслятором. В графе 0-транслятор не является ничьим ресурсом, но сам программирует и конфигурирует другие трансляторы. Смешанные трансляторы могут по внешним входам получать некоторую информацию извне, а по другим – команды на передачу этой информации по графу актуализации. Сама ИУС обычно является смешанным транслятором.

Любой транслятор является программируемым (0-программируемым). Если транслятор конфигурируемый, то он одновременно является и программируемым, и конфигурируемым. Конфигурируемый транслятор может менять реакции, но без подачи программирующей команды эти реакции не будут генерироваться.

Если в актуализации участвует большое число трансляторов, то можно вводить иерархию, выделяя подграфы. Граф актуализации критически зависит от иерархических уровней. Информация графа включает в себя трансляторы, потоки данных и все их команды. При введении иерархии меняется состав трансляторов и, что важно, состав их команд. Фактически команда останется прежней, но ее влияние на новый транслятор будет другим – команда может поменять тип. Начиная с самого верхнего представления ИУС как одного транслятора, можно опускаться на более детальные уровни, на которых будут появляться новые трансляторы и команды. При этом будут детализироваться не только трансляторы, но и их ресурсы. Так можно опускаться вплоть до логических элементов и электрических сигналов на их входных и выходных линиях.

Модель позволяет неограниченно детализировать граф актуализации. При этом большинству разработчиков не нужен уровень детализации одного порядка для разных фрагментов графа: разработчики инструментального программного обеспечения хорошо представляют фазу разработки, разработчики встроенного программного обеспечения – подграф программных трансляторов фазы исполнения. Чтобы иметь одинаковое (по уровню) представление обо всех частях графа актуализации, необходимо участвовать в разработке всех его трансляторов. Для современных ИУС это большая редкость, поэтому на практике граф актуализации представляется фрагментарно и отображает лишь то, что важно для разработчика, лишь то, на что он может повлиять. Формально создание полного графа актуализации затруднительно, поскольку в граф входят трансляторы, разработанные сторонними фирмами-разработчиками – электронные компоненты, библиотеки функций и др. Даже если известны их входные и выходные потоки, то внутренние связи являются закрытой информацией.

Фазы актуализации

Разделим процесс актуализации на 2 фазы и раскрасим вершины графа (трансляторы) 2-мя цветами в соответствии с принадлежностью одной из фаз:

- фаза разработки (design-time, DT) — желтая;
- фаза исполнения (run-time, RT) — зеленая.

Обе фазы участвуют в процессе актуализации ЦП, но по-разному. Фаза разработки совпадает с этапом разработки ИУС. Все формы представления ЦП на этой фазе предназначены для подготовки к его исполнению. Ни одна из форм представления на фазе разработки не пригодна для ее исполнения ИУС, как транслятором. Трансляторы фазы исполнения функционируют во время работы ИУС при каждом ее запуске. Основное отличие трансляторов этих двух фаз заключается во времени их работы. Работа RT-трансляторов – это работа системы. Работа DT-трансляторов происходит вне ИУС и до ее готовности. Очевидно, что в графе актуализации ЦП сначала попадает в фазу разработки, а затем в фазу исполнения.

DT-трансляторы преобразуют целевой алгоритм в форму, которая исполняется ИУС. Уже давно используется технология, по которой эта форма получается один раз, а исполняется много раз. Чтобы каждый раз не проделывать работу фазы разработки, нужны энергонезависимые хранилища информации. Без этих хранилищ полученную форму ЦП невозможно было бы сохранить для повторного использования, и при каждом старте нужно было бы программировать ИУС заново. Для большинства окружающих нас вычислительных систем это считается очевидной нормой, но раньше это было не так, и сейчас для некоторых видов ИУС (Evaluation Boards) это не так.

Разделение процесса актуализации на фазы позволяет дать новую трактовку широко распространенным в области разработки ИУС определениям. Одни трансляторы актуализируют целевой алгоритм внутри одной фазы, а другие переводят между фазами. Транслятор, переводящий потоки в фазу исполнения, называется *компилятором*. Т.е. компилятор – это тот транслятор, чьи выходные потоки могут быть актуализированы ИУС. Связанный подграф, состоящий из DT-трансляторов и заканчивающийся компилятором, называется *инструментальной цепочкой*. У ИУС, если она, например, многопроцессорная система, может быть несколько инструментальных цепочек. Для каждого из ее процессоров может быть своя инструментальная цепочка. *Программное обеспечение ИУС* (ПО ИУС) – это совокупность выходных потоков всех компиляторов, т.е. все потоки, приходящие в фазу исполнения. Традиционно под программным обеспечением понимается несколько другое, то, что создают программисты. Программисты как трансляторы вынесены за рамки графа актуализации, поэтому генерируемая ими форма представления ЦА не попадает непосредственно в фазу исполнения.

Организация фазы исполнения ИУС

Каждый RT-транслятор обладает специфической архитектурой и содержит набор команд. Изучать и использовать язык каждого транслятора для актуализации ЦП неэффективно. Для повышения эффективности программирования вводятся специальные трансляторы, через которые осуществляется программирование других трансляторов. Другими словами, на фазе исполнения выделяют такие трансляторы, чтобы остальные RT-трансляторы являлись их ресурсами. Назовем такие трансляторы *базовыми (БТ)*. Выгода от использования БТ очевидна: программирование их ресурсов выполняется только с помощью языков БТ, а их в системе гораздо меньше, чем ресурсов. Наибольшее распространение получили следующие БТ: микропроцессоры, микроконтроллеры и программируемые логические интегральные схемы (ПЛИС).

Рассмотрим задачу программирования произвольного транслятора Т. Это можно сделать через его входной поток (рис. 2, а, поток А). Если транслятор Т доступен, как ресурс БТ, то его можно запрограммировать с помощью команд БТ (рис. 2, б, потоки Б–В). Формируется новая задача: как запрограммировать БТ, чтобы генерируемый им поток В оказался аналогичным потоку А. Можно ввести промежуточный транслятор Д.

Его задачей будет обеспечение актуализации тех функций транслятора Т, которые требуются для актуализации ЦП (рис. 2, в, потоки Г-Е-Ж). Такой транслятор назовем *драйвером*.

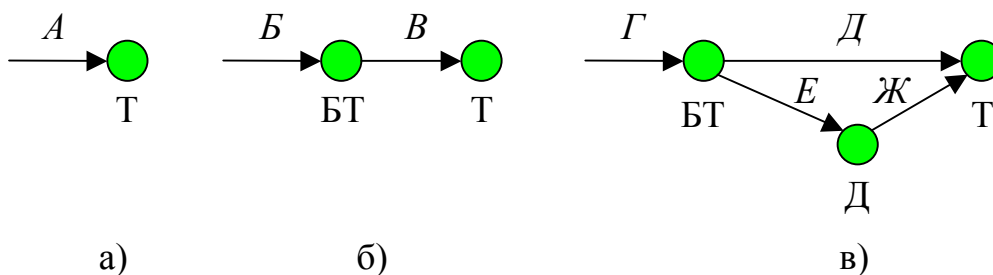


Рис. 2. Способы программирования трансляторов

Например, последовательный канал можно программировать через регистры специального назначения, поддерживаемые микроконтроллером, а можно через драйвер последовательного канала. Вызов функций драйвера обеспечивает микроконтроллер, поэтому драйвер является ресурсом микроконтроллера. В некоторых случаях при наличии драйвера непосредственное программирование ресурса (рис. 2, е, потоки Г-Д) может быть запрещено, и единственным вариантом останется программирование через драйвер (рис. 2, в, потоки Г-Е-Ж).

Введение драйверов усложняет структуру графа актуализации, но для программирования ресурса нужно определить лишь первый программный поток. В первом случае это будет поток А, во втором – В, а в третьем – Г. При этом поток Г через БТ может привести к программированию через поток Е и/или Д. В первом случае нужно реализовать функции для транслятора Т в его входном потоке, во втором – в потоке БТ, а в третьем случае эти функции уже реализованы драйвером и в потоке Г их нужно запустить на исполнение. Тем самым упрощается программный поток для БТ. Использование термина *драйвер* интуитивно более понятно для программирования микроконтроллеров и микропроцессоров. Для ПЛИС аналогом драйверов являются *мегафункции*. Упрощение входных потоков, которые нужно задать разработчикам ИУС, привело к увеличению сложности ее структуры. До тех пор, пока эта сложность не превысит простоту программирования потоков, будут пользоваться этим методом.

Таким образом, для упрощения программирования ресурсов ИУС вводится сеть драйверов, актуализирующая частные задачи, входящие в актуализацию ЦП. Это является одним из способов организации фазы исполнения графа актуализации. Дальнейшим развитием драйверов является появление устойчивых комбинаций драйверов, предоставляющих не только доступ к ресурсам ИУС, но и управляющих вычислительным процессом. Сначала это были библиотеки функций, затем они преобразовались в *операционные системы* (ОС). Программирование для ПК достигло уже такого уровня, что число функций, которые он может выполнять, столь велико, что программисты специализируются лишь на некоторых типах драйверов: разработке баз данных, WEB-интерфейсов, пользовательских приложений, приложений клиент-сервер и др. Программированием процессора ПК как БТ уже никто не занимается, потому что реализация функций, которые целесообразно решать с учетом вычислительной мощности современных ПК, слишком сложна.

Для упрощения актуализации ЦП на фазу исполнения могут быть добавлены не только драйверы, но и новые БТ. У них может быть новая архитектура, новые входные команды, более предпочтительные для актуализации ЦП. Программно реализованный БТ будем называть *виртуальной машиной* (ВМ). Если доступ к ресурсам обеспечивается с помощью драйверов, то они скрывают особенности ресурсов: ресурсы могут быть разными (разные модели устройств), а функции работы с ними будут одинаковыми. Программирование получается аппаратно-независимым, но зависимым от драйверов.

Программирование ВМ аналогичным образом является независимым от используемого БТ, но зависимым от самой ВМ (см. рис. 3). В программировании под ПК ВМ применяются гораздо шире, чем в ИУС. Perl, Java, Smalltalk, VBScript широко распространены, и разработчики ПО для этих ВМ востребованы.

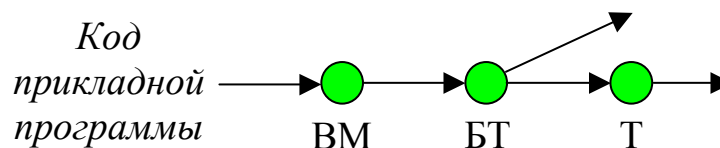


Рис. 3. Добавление виртуальной машины на фазу исполнения

Ни один транслятор не функционирует мгновенно, и это влияет на производительность фазы исполнения: чем больше в ней будет трансляторов, тем меньше будет производительность ИУС. Добавление драйверов и ВМ с целью организации вычислительного процесса понижает производительность ИУС. Добавление драйверов вносит незначительно понижение производительности, потому что драйверы добавляются между БТ и ресурсами: БТ программируется с первоначальной производительностью, а ресурсы – уже с пониженной. Добавление ВМ вносит значительное понижение производительности, потому что исходный БТ сам становится ресурсом и его программирование замедляется, а, значит, замедляется доступ ко всем его ресурсам.

Заключение

В статье представлено введение в модель актуализации. В нее входят трансляторы, функционирующие на разных фазах процесса актуализации и реализованные программно и аппаратно. Из-за ограничения объема статьи не были описаны следующие фрагменты модели:

- организация взаимодействия программных трансляторов;
- подсистема прерываний;
- системное и прикладное программное обеспечение;
- организация программного управления в соответствии с классификацией [5];
- способы построения графа актуализации.

На текущий момент в модели актуализации еще есть нерешенные вопросы, и получение ответов на них является важнейшей задачей. Для проверки модели были созданы средства автоматизированного построения графа актуализации по существующей ИУС. Практика показала, что графы современных систем слишком сложны для визуального восприятия: они насчитывают 50 и более трансляторов только на фазе исполнения и являются непланарными. Планируется автоматическая обработка графа и получение его характеристик.

Литература

1. Lee E. What's Ahead for Embedded Software? // Computer. 2000. № 9. P. 19–28.
2. Lee E. Absolutely Positively on Time: What Would It Take? // Computer. 2005. № 7. P. 85–87.
3. Брукс Ф. Мифический человеко-месяц или как создаются программные системы. М.: Символ-Плюс, 2000.
4. Lavagno. L, Sangiovanni-Vincentelli A., Sentovich E. Models of Computation for Embedded System Design. // System-level synthesis, 1999. P. 45–102.
5. Ковязин Р.Р., Платунов А.Е. Программирование микроконтроллерных систем // Электронные компоненты. 2003. №4. С. 65–70.

ПОИСК ПРОГРАММНЫХ КОМПОНЕНТ ДЛЯ ПОВТОРНОГО ИСПОЛЬЗОВАНИЯ С ПОМОЩЬЮ АВТОМАТИЗИРОВАННЫХ СИСТЕМ ДОКАЗАТЕЛЬСТВ ТЕОРЕМ

О.Г. Шедько, В.В. Окулевич

Настоящая работа рассматривает вопросы, связанные с поиском программных компонент при разработке программного обеспечения на базе процесса повторного использования. В частности, авторы предлагают вариант решения ряда проблем, возникающих при аппроксимационном поиске методом Мили.

Введение

На сегодня одной из самых больших проблем в области разработки программного обеспечения (ПО) на базе повторного использования (ПИ) программных компонентов является поиск подходящих объектов в соответствующих библиотеках, а также выбор наиболее оптимального компонента (требующего минимальной адаптации под новую задачу) среди этих кандидатов [3]. По мере роста числа компонентов в библиотеке время и стоимость поиска увеличиваются, а возможность нахождения кандидата вручную уменьшается.

Решение проблемы состоит в создании автоматических средств поиска. Однако далеко не каждый известный метод поиска предполагает наличие базы для автоматизации, которая включает в себя некий стандарт для описания, хранения и сравнения готовых компонент, а также критерии и правила, по которым определяются отношения между составляющими библиотеки.

Использование топологических аппроксимационных методов поиска и выбора компонент, а также автоматизированных систем доказательств теорем, поддерживающих эти методы, позволяют решить эти вопросы [1]. Однако применение указанных методов связано с рядом ограничений, снижающих эффективность поиска.

В настоящей работе авторы предлагают решение двух проблем, характерных для методов данного класса:

1. невозможность нахождения компонент повторного использования (КПИ) при низкой степени детализации требований в описании ключевого запроса на поиск;
2. невозможность нахождения КПИ при полном несовпадении какого-либо из требуемых свойств у искомого компонента и кандидата на ПИ.

Поиск программных компонент для повторного использования с помощью автоматизированных систем доказательств теорем

Рассмотрим подробнее топологические аппроксимационные методы поиска и выбора компонент. Основными отличительными особенностями указанных методов является:

1. описание компонент и отношений между ними для хранения, поиска и выполнения запросов к базе производится в виде формальных спецификаций;
2. база компонент предполагает иерархическое упорядочивание элементов в ней;
3. сравнение компонент при поиске производится при помощи вычисления некоторых дистанций, определяющих степень их сходства или различия;
4. описание, поиск и выбор компонент производится при помощи автоматизированных систем доказательств теорем, реализующих логику метода поиска.

Топологические аппроксимационные алгоритмы поиска могут использовать как структурные, так и функциональные описания компонент.

С точки зрения практического применения методов поиска для повторного использования компонент наибольший интерес представляют методы со следующими характеристиками:

1. наличие возможностей как для функционального, так и для структурного описания компонент существенно расширяет область применения метода;
2. наличие возможностей для самостоятельного пополнения библиотеки КПИ, а также определения новых иерархических отношений между составляющими библиотеки дает возможность самостоятельно создавать библиотеки из новых компонентов и реализовывать поиск и сравнение в них;
3. минимизация использования экспертных оценок при определении дистанций между компонентами позволяет при выборе исключить ряд ошибок связанных с влиянием человеческого фактора.

По результатам проведенного авторами анализа среди топологических аппроксимационных методов поиска компонент данным требованиям наиболее полно соответствует метод Мили [1, 4]. Метод был разработан в 1997 Мили, Джилани, Дешарнаис и др. и поддерживается системой автоматизированного доказательства теорем (так называемым прувером) «Оттер» [2]. Для сравнения и оценки различий между запрашиваемым компонентом и кандидатами на повторное использование здесь используются четыре параметра, определяющих функциональную дистанцию, – функциональный консенсус, функциональное различие, уточняющее расстояние, уточняющее отношение. Первый параметр показывает, сколько общего есть у двух компонент, второй – сколько необходимо добавить, чтобы из первого компонента получить второй, третий – какое количество irrelevantных функций содержат эти два компонента по отношению друг к другу, четвертый параметр объединяет все предыдущие.

Метод Мили обладает рядом существенных преимуществ перед другими топологическими аппроксимационными методами. Однако его использование сопряжено с рядом трудностей, порожденных как особенностями самого метода, так и инструмента для его автоматической поддержки. Остановимся на этих трудностях подробнее.

Рассмотрим для этого пример сравнения двух компонент и ключа на основе вычисления функционального консенсуса.

Пример 1. Пусть мы имеем библиотеку планировщиков задач. Описание каждого планировщика в библиотеке задается рядом входных и выходных характеристик. Ко входным характеристикам относятся: алгоритм планирования (FIFO, RR, RM, ADAPT), периодичность поступления задач в систему (periodic, aperiodic, full) и прерываемость задач (preemptive, interrupt). К выходным относится количество задач, которые может обработать планировщик в единицу времени (small, medium, large, super).

Для входных характеристик (X) определены следующие отношения:

```
% task periodicity
(all x (periodic(x) -> full(x))).1
(all x (aperiodic(x) -> full(x))).
```

```
%possibility to interrupt task
(all x (preemptive(x) -> interrupt(x))).
```

```
%scheduling algorithms
(all x (fifo(x) -> methods(x))).
(all x (rr(x) -> methods(x))).
(all x (adapt(x) -> methods(x))).
```

¹ Данная запись означает, что для всех значений X множество значений periodic(x) является подмножеством full(x) или full(x) включает в себя periodic(x)

$$(all\ x\ (rm(x) \rightarrow methods(x))). \quad (1)$$

Для выходных характеристик (Y) определены следующие отношения:

$$\begin{aligned} &\% \text{ task quantity} \\ &(all\ y\ (super(y) \rightarrow large(y))).^2 \\ &(all\ y\ (large(y) \rightarrow medium(y))). \\ &(all\ y\ (medium(y) \rightarrow small(y))). \end{aligned} \quad (2)$$

Пусть в библиотеке находится 2 алгоритма, которые определены следующим образом:

$$\begin{aligned} A1(x, y) &= rr(x) \& preemptive(x) \& full(x) \& small(y). \\ A2(x, y) &= rr(x) \& preemptive(x) \& full(x) \& medium(y). \end{aligned} \quad (3)$$

Предположим, что искомый компонент задан с помощью ключа поиска следующим образом:

$$key(x, y) = rr(x) \& preemptive(x) \& periodic(x) \& large(y). \quad (4)$$

После вычисления функционального консенсуса($\emptyset$) между ключом и компонентами мы получим следующее:

$$\emptyset(key, A1) \rightarrow \emptyset(key, A2) \quad (5)$$

Из этого следует, что A2 является более подходящим кандидатом на повторное использование для создания данного ключа.

Теперь на примере 1 рассмотрим недостатки метода.

Предположим, что разработчик при поиске компонента задал лишь только существенные для него характеристики. Это значит, что степень детализации требований в описании ключевого запроса на поиск оказалась ниже степени детализации описания компонентов в библиотеке. Это довольно часто встречающаяся ситуация, поскольку для нового компонента ряд свойств старого может быть абсолютно irrelevantен. Например, разработчик ищет планировщик, обрабатывающий большое количество периодических прерываемых задач с любым алгоритмом планирования. Тогда ключ будет выглядеть так:

$$key(x, y) = preemptive(x) \& periodic(x) \& large(y). \quad (6)$$

Однако здесь возникают две проблемы. Во-первых, прувер «Оттер» не имеет базовой конструкции для описания irrelevantных функций, а также не может сравнивать две спецификации при отсутствии в одной из них описания какого-либо свойства, присутствующего в другой. Во-вторых, метод Мили поддерживает понятие irrelevantности свойств компонента только при условии обязательного наличия пересечения компонента и ключа (рис.1). Для ряда случаев, когда компонент может быть повторно использован после адаптации, метод, тем не менее, выдает в качестве результата пустое множество кандидатов.

Для решения указанных проблем мы предлагаем ввести понятие irrelevantного множества следующим образом.

² Для выходных параметров операция включения имеет обратный смысл. Так $super(y) \rightarrow large(y)$ означает, что множество $super(y)$ включает в себя $large(y)$.

Определение 1. Пусть для некоторого функционального свойства A доменной области определен ряд аксиом, задающий отношения между множествами его значений. Тогда каждый элемент иррелевантного множества будет являться подмножеством любого множества значений, определенного для свойства A .

Введение иррелевантного множества решает проблему различной степени детализации между ключевым запросом и описанием компонентов, а также проблему ограничений на понятие иррелевантности метода Мили.

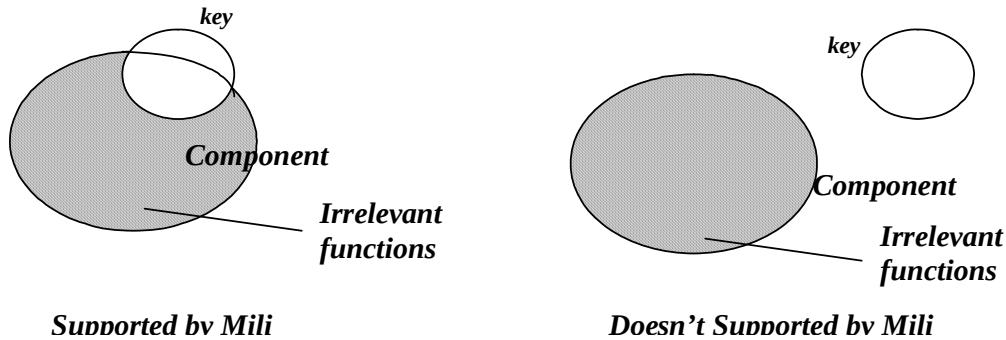


Рис. 1. Поддержка понятия иррелевантности методом Мили

Особенность метода, связанная с тем, что для успешного нахождения КПИ требуется наличие пересечений во всех свойствах компонента и ключевого запроса порождает еще одну проблему. Актуальность данного требования не вызывает сомнения, если компонент в библиотеке задан лишь одним свойством. Однако, как правило, описание компонента задается рядом его свойств. В этом случае метод может отсеять ряд компонентов, посчитав их не пригодными для использования после адаптации. Это существенно сужает множество кандидатов на ПИ, а в ряде случаев дает пустое множество кандидатов. Поясним это на примере 1.

Пусть ключевой запрос задан с помощью формулы (7):

$$\text{key}(x, y) = \text{rm}(x) \ \& \ \text{preemptive}(x) \ \& \ \text{periodic}(x) \ \& \ \text{large}(y). \quad (7)$$

При этом мы видим, что пересечение в области свойства, характеризующего алгоритм планирования, в ключе и описании планировщиков отсутствует. Таким образом, множество кандидатов для ПИ будет пустым.

Однако очевидно, что в области других свойств подобное пересечение есть. Если предположить, что все свойства искомого компонента в данном случае имеют для разработчика одинаковую важность, то подобное сужение множества результатов является существенным недостатком.

Для решения данной проблемы предлагается использовать механизм генерализации и иррелевантные множества. Под генерализацией в данном случае имеется в виду механизм обобщения ключевого запроса в области отсутствия пересечения с компонентами при помощи иррелевантного множества. Рассмотрим решение проблемы для ключа в формуле (7).

Введем иррелевантное множество **nill_alg** для свойства «алгоритм планирования»:

$$\begin{aligned} & \% \text{scheduling algorithms} \\ & (\text{all } x \ (\text{nill_alg}(x) \rightarrow \text{fifo}(x))). \\ & (\text{all } x \ (\text{nill_alg}(x) \rightarrow \text{rr}(x))). \\ & (\text{all } x \ (\text{nill_alg}(x) \rightarrow \text{adapt}(x))). \\ & (\text{all } x \ (\text{nill_alg}(x) \rightarrow \text{rm}(x))). \end{aligned} \quad (8)$$

Произведем генерализацию ключевого запроса:

$$\text{key}(x, y) = \text{nill_alg}(x) \ \& \ \text{preemptive}(x) \ \& \ \text{periodic}(x) \ \& \ \text{large}(y). \quad (9)$$

Выполним вычисление функционального консенсуса для нового ключа и компонент библиотеки при помощи прувера и найдем кандидатов для ПИ.

Введение механизма генерализации, с одной стороны, повышает результативность поиска, с другой стороны – вносит дополнительные временные затраты, связанные со следующими моментами:

1. необходимо переупорядочивание компонент в библиотеке относительно нового запроса;
2. требуется осуществить новый поиск в библиотеке;
3. произведение повторных генерализаций в случае неочевидности области отсутствия пересечения.

Часть данных затрат можно существенно сократить следующим путем.

1. Хранение графов отношений между компонентами для генерализаций свойств компонент, вычисленных ранее. Это позволит в ряде случаев избежать переупорядочивания описаний.
2. Введение коэффициентов важности свойств в описании компонентов. Это позволит установить однозначный приоритет порядка генерализации в случае неочевидности области отсутствия пересечения, а также выполнять оценку, есть ли смысл в проведении последующих генерализаций.

Заключение

Подводя итоги, еще раз отметим, что авторами были предложены пути решения проблем, приводящих к снижению результативности поиска по методу Мили с помощью прувера «Оттер». Использование предложенных способов и приемов позволяет снять ограничения метода и прувера и тем самым повышает эффективность процесса поиска и адаптации компонент для их ПИ.

Литература

1. Jilani L.L., Mili R., Mili A. and etc Retrieving Software Components that Minimize Adaptation Effort. – Tunisia-USA-Canada, 1997.
2. McCune W. Otter3.3 Reference Manual // Argonne National Laboratory, Technical Memorandum №263, 2003.
3. Mili H., Mili F., Mili A. Reusing Software: Issues and Research Direction. USA. Canada, 1995.
4. Mili R., Mili A., Mittermeir R. T. Storing and Retrieving Software Components: A Refinement Based System. // IEEE Transactions On Software Engineering, Vol. 23, № 7, July 1997.

СРЕДСТВА ТЕСТИРОВАНИЯ ПЕЧАТНЫХ ПЛАТ С ИСПОЛЬЗОВАНИЕМ ТЕХНОЛОГИИ JTAG

В.И. Скорубский, Д.В. Овчаров, Д.А. Макаров

Рассматривается технология автоматического тестирования печатных плат (PCB) с использованием граничного сканирования и JTAG-интерфейса с применением аппаратной поддержки исполнения контроллером. Определены условия для повышения разрешающей способности системы тестирования.

Введение

JTAG-технология [1] контроля печатных плат для сложной цифровой аппаратуры с установленными ПЛИС, быстродействующими контроллерами и микропроцессорами – единственно доступная в настоящее время. Несмотря на распространение и наличие большого числа публикаций по этой теме, сохраняется определенная дистанция между разработчиком и фирмами-изготовителями схем и тестового оборудования. Таким образом, приходится практически заново создавать средства тестирования.

В работе предлагается обзор прикладной системы, поддерживающей технологию JTAG. Обзоры [2–5] содержат рекомендации известных фирм Altera, Xilens, Atmel, TI по применению технологии к конкретным изделиям этих фирм – главным образом, при внутрисхемном программировании отдельных модулей. В целом, отсутствуют общедоступные средства и какая-либо методика организации тестирования и выбора тестов при искусственных ограничениях – доступен весьма ограниченный список команд тестирования, не всегда можно получить описание микросхем в BSDL- файлах.

Фирма Altera предлагает Jam- плееры для интерпретации Jam-программ записи ПЛИС, формируемых системой MAX+.на PC или контроллером с архитектурой MCS51. Во втором случае требуется дополнительное преобразование Jam-программ в программу виртуальной машины. Интерпретаторы Jam-текстов приспособлены только для программирования ПЛИС, и, несмотря на то, что тексты исходных C-файлов доступны, вмешаться или адаптировать их к приложениям тестирования не представляется возможным.

Тем не менее, архитектура системы тестирования с плеером, реализованная, например, в эмуляторах MSP430 фирмы TI, может быть успешно использована в собственных системах тестирования. В работе рассматривается архитектура и особенности реализации системы тестирования, не ограниченная только элементами одной фирмы.

Технология JTAG-тестирования

На рис. 1 приведена общая схема системы тестирования с использованием специализированного контроллера Player. Здесь может быть использован любой контроллер, имеющий достаточные ресурсы – не менее 8 кБ Flash, 2 кБ Ram, интерфейс rs232 и JTAG-порт. В частном случае для контроля исполнения тестирования в автоматическом режиме и автономного использования могут быть полезны локальные средства индикации (ЖКИ) и оперативного ручного управления (функциональная клавиатура).

Элементы архитектуры системы тестирования: PCB – печатная плата, Player – исполнение JTAG-команд и связь с Монитором на ПК, Монитор – система исполнения тестов и формирования базы данных txt на основе BSDL-файлов и тестируемых цепей Pins. В отличие от фирменных решений, приходится отказаться от прямой интерпретации Jam-файлов, что позволяет сократить объем программ плеера и увеличить скорость обмена данными.

Существенная функция Player'a – управление автоматом ТАР Jtag-интерфейса. Интерпретация конечного автомата – одна из распространенных задач при работе с моделями алгоритмов, обобщение этой задачи – выбор кратчайших путей на графе.

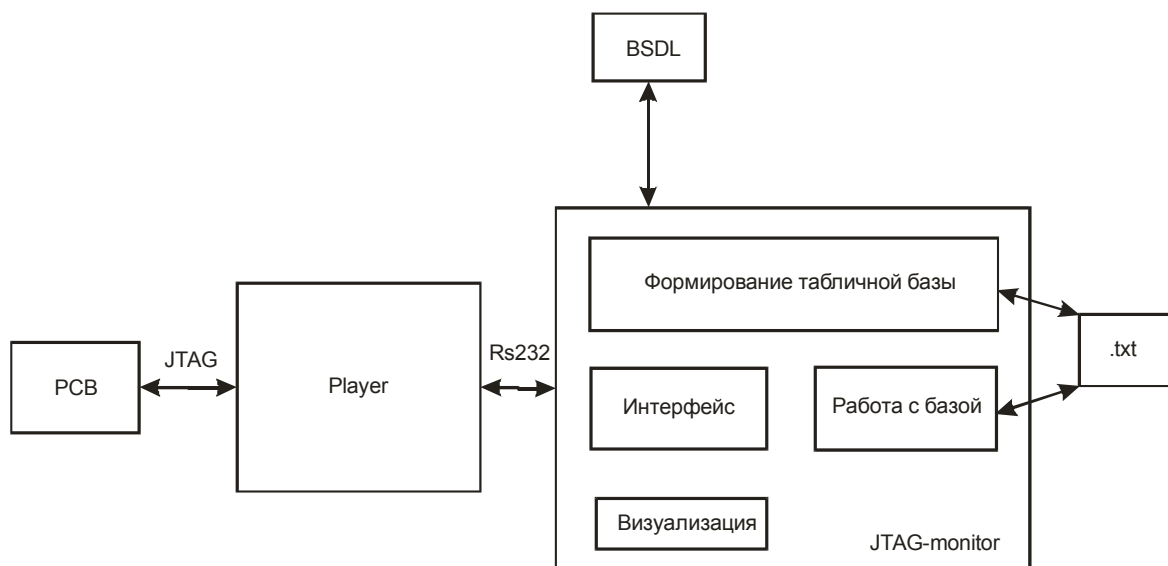


Рис.1. Схема тестирования в JTAG-технологии

Используя символическое кодирование ребер графа, приходим к задаче поиска кратчайшего пути в КА, где вершины графа – состояния автомата, а кодирующие символы – условия перехода между состояниями.

Алгоритм выбора кратчайшего пути

Традиционное решение задачи выбора кратчайшего пути требует значительного перебора – например, распространением числовой волны. О важности этой задачи свидетельствует то, что к заслугам Дейкстры [5] относят решение задачи выбора кратчайшего маршрута. Задача упрощается, если информация о кратчайших маршрутах получена предварительным анализом одним из известных алгоритмов или простым перебором, а в реальном времени выбирается маршрут между двумя заданными точками на ее основе.

Задача может быть сведена к выбору кратчайшего пути в закодированном графе, который интерпретируется конечным автоматом в некотором алфавите. Эффективный метод решения задачи для конечного автомата ТАР применяется в Player51 фирмы Altera

Пусть автомат задан матрицей переходов $N \times N$, где N – число состояний. Предположим нетривиальный случай, когда автомат детерминирован и полностью определен и число состояний $Q = \{q_1, \dots, q_N\}$ больше числа символов входного алфавита $\Sigma = \{a, b, \dots\}$. Тогда матрица переходов содержит пустые клетки.

В тривиальном случае все клетки заняты, и решение тривиально – для любой пары состояний существует кратчайший путь из одного условия. Если матрица содержит пустые клетки, то выберем первую строку матрицы и первую пустую клетку в ней – пусть с индексами (i, j) . Пустая клетка обозначает факт, когда необходимо найти кратчайший путь, соединяющий состояния (q_i, q_j) . Найдем такой кратчайший путь любым методом (можно визуальным просмотром или перебором). Тогда в клетку (i, j) запишем выделенное условие перехода в следующее состояние $a \in \Sigma$ кратчайшего пути из q_i (пусть в $q_m/m \neq i, j$). Далее рассматривается кратчайший путь $E_{q_m q_j}$. Если клетка (m, j)

не пустая, то алгоритм доопределения матрицы кратчайших путей завершается. В противном случае заполняется пустая клетка (m, j) в строке m аналогичным образом. Таким образом, за конечное (не более $N-2$) число шагов будут заполнены все клетки матрицы переходов выделенными условиями.

В дальнейшем любой кратчайший путь E_{qi}, qj может быть выбран простым алгоритмом, состоящим из одинаковых шагов.

1. Выбрать клетку матрицы (i, j) .
2. Если условие в клетке не отмечено, то путь найден и заканчивается этим условием.
3. В противном случае перейти по выделенному условию α в следующее состояние, используя условие, не выделенное в строке i .
4. Продолжить движение по кратчайшему пути до завершения его в п.2.

Методика построения матрицы кратчайших путей является доказательством конечности и корректности алгоритма поиска кратчайшего пути в графе автомата.

Пример формирования матрицы для детерминированного КА, заданного графом и таблицей переходов

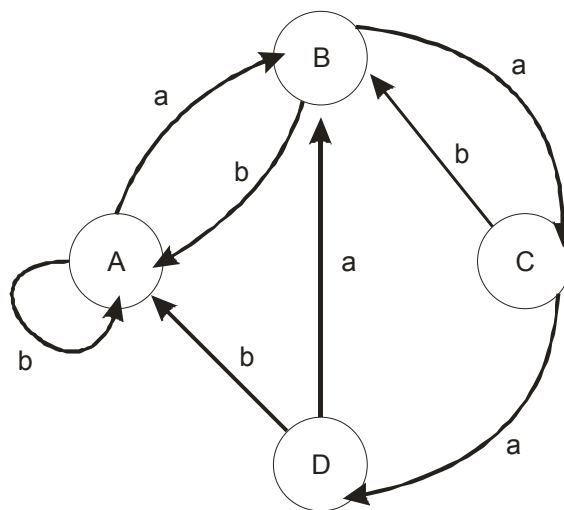


Рис. 2. Граф КА

Q_i	Σ	Q_j
A	a	B
	b	A
B	a	C
	b	A
C	a	D
	b	B
D	a	B
	b	A

Табл. 1. Таблица переходов КА

В матрице выделяются заданные условия перехода. В данном случае первая пустая клетка в первой строке матрицы (A, C) . Соответствующий кратчайший путь из A в C проходит через вершину B, и первое условие 'а' перехода из A в B запишем в клетку (A, C) . Условие перехода из B в C уже присутствует в матрице, и первый проход с доопределением кратчайшего пути заканчивается. Следующей рассматривается пустая клетка (A, D) , и для нее кратчайший путь $A \rightarrow B \rightarrow C \rightarrow D$. При этом условием 'а' заполняется пустая клетка (A, D) . Остальные клетки на этом пути уже заполнены.

	A	B	C	D
A	<u>b</u>	<u>a</u>		
B	<u>b</u>		<u>a</u>	
C		<u>b</u>		<u>a</u>
D	<u>b</u>	<u>a</u>		
	A	B	C	D

Таблица 2. Исходная матрица переходов

Аналогичным способом заполняем клетки кратчайших путей $B \rightarrow A \rightarrow B$, $B \rightarrow C \rightarrow D$, $C \rightarrow B \rightarrow A$, $C \rightarrow B \rightarrow C$. После заполнения матрица имеет вид.

	A	B	C	D
A	<u>b</u>	<u>a</u>	a	<u>a</u>
B	<u>b</u>	a	<u>a</u>	<u>a</u>
C	a	<u>b</u>	a	<u>a</u>
D	<u>b</u>	<u>a</u>	a	<u>a</u>
	A	B	C	D

Таблица 3. Матрица кратчайших путей

Для выбора кратчайшего пути между двумя заданными вершинами графа по матрице кратчайших путей следует двигаться по заданным условиям последовательно. Например, из A в D попадаем, выбирая условие 'a' в клетке (A,D), затем 'a' в клетке (B,D) и завершающее выделенное условие 'a' в клетке (C,D).

Алгоритм можно применить для выбора кратчайшего пути в реальном времени на взвешенном графе – частный случай задачи коммивояжера.

В TAP условия принимают только значения 0, 1 для сигнала TMS в JTAG-интерфейсе, для состояний используется естественная последовательная нумерация (многозначное кодирование) и матрица кратчайших путей задана 16 двоичными кодами. Для перехода по отмеченным вершинам используется таблица переходов.

Особенности тестирования в реальном времени

Тестирование внешних цепей для микросхемы осуществляется TAP-командой EXTEST. Предварительно командой SAMPLE/Preload загружается вектор, который переводит двунаправленные I/O выходы контактов микросхемы в Z-состояние или устанавливает выходы в контрольные состояния.

Таким образом, разрешающая способность теста определяется задержкой алгоритма выполнения функций SAMPLE и EXTEST, представленного на рис. 3, где N – длина теста в битах, t_s – задержка байта в последовательном канале, m – длина команды в битах, t_j – задержка бита в JTAG.

Диагностика, в которой используется эталонный вектор для сравнения, может совмещаться с выполнением команды EXTEST, как это предусмотрено в Jam-тестах. Вместе с тем для сокращения времени на сохранение реакции в BSR (TAP – состояние Capture) целесообразно выполнять диагностику и контроль в виде отдельной функции и возлагать ее на монитор в ПК. Точно также из цикла тестирования исключается загрузка векторов из ПК.

Таким образом, приведена подробная последовательность работы TAP, управляемая командами Play и совмещенная с подготовительными операциями.



Рис. 3. Схема алгоритма выполнения теста

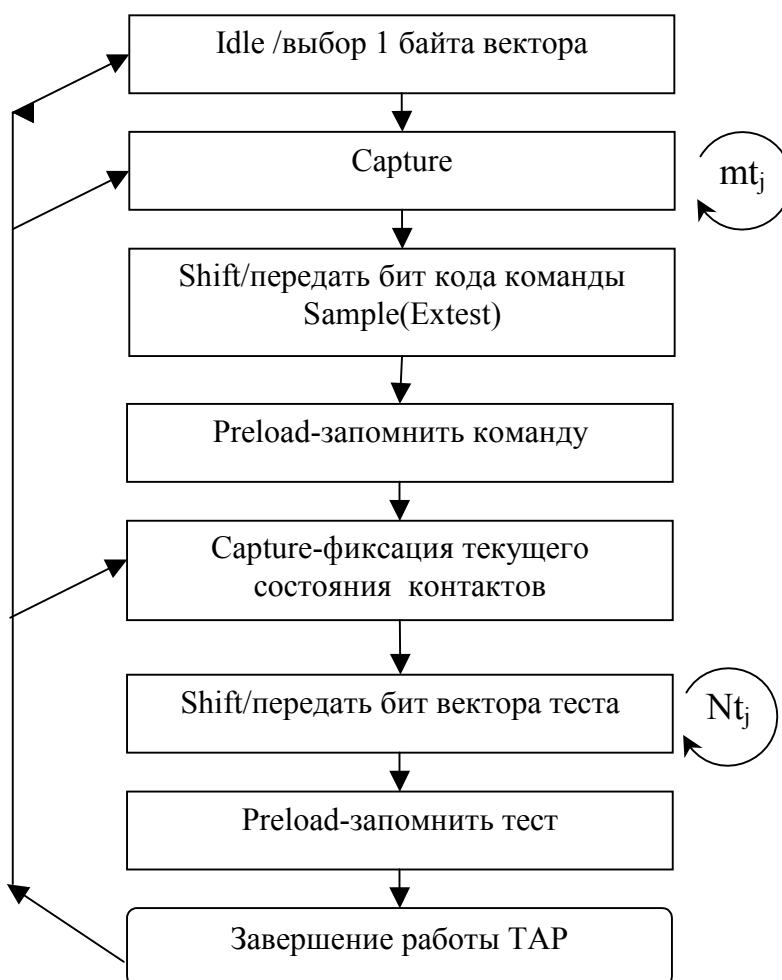


Рис. 4. Схема исполнения команд Sample, Extest

Для тестирования соединений РС в статике при исполнении таких функций, как обращение к памяти или передача данных, последовательность операций ТАР-контроллера выполняется в диалоговом пошаговом режиме и совмещается с формированием кодов тестирования. Функциональный тест может быть представлен микрокомандами с использованием функциональных обозначений тестируемых линий – например, операции записи и чтения во внешней RAM, инициируемые ПЛИС:

AM7-AM0 = 0; DM15-DM00=0хaaaa; cs=0; wr=0; oe=1;

AM7-AM0 = 0; DM15-DM00=zzzz; cs=0; wr=1; oe=0;

Модуль RAM не имеет Jtag цепочки и тестируется ПЛИС косвенно.

В режиме многократного повторного тестирования ТАР проходит состояния Capture, Shift, Preload, пребывает $N+m$ тактов в состоянии Shift (длина теста N – сотни и тысячи бит). Объем теста зависит от разрешения диагностики и определяется следующими уровнями:

- контактами отдельных схем, где сначала все доступные контакты устанавливаются в Z-состояние командой HIGHZ, затем группы контактов каждой схемы проверяются на наличие контакта;
- электрические цепи на наличие соединения (для n цепей формируются $2N$ бит теста – передача и прием 0, 1);
- короткие замыкания цепей (в [7] объем теста оценивается как $[\log(2(2N+2))N]$ -битовых векторов).

В реальном времени тестирующий вектор формируется с учетом динамики работы всей системы. Если контакт работает на вход In, то он может принимать различные состояния внешних схем, что затрудняет диагностику. Если контакт выхода Out, то состояние управляющего бита BSR согласуется с состоянием Z принимающего контакта, что невозможно, если этот контакт не доступен в BSR. В этом случае необходима команда Intest, позволяющая отключать внешние схемы, не связанные Jtag. К таким схемам относятся внешние синхронизаторы, формирователи. Для контроля и диагностики одиночных неисправностей в цепях связи Jtag-технология достаточна, но, естественно, диагностика множественных неисправностей [7] может быть затруднена или невозможна.

Заключение

Рассмотрена схема тестирования с JTAG-интерфейсом. Основными этапами, задержки которых влияют на разрешающую способность тестов, являются этапы исполнения автомата ТАР. Предлагается алгоритм доопределения матрицы переходов конечного автомата и выбора кратчайшего пути между двумя состояниями в ТАР. Разработанная программа в С++ используется для тестирования сложных печатных плат, содержащих несколько ПЛИС EPM7xxx фирмы Altera.

Литература

1. Платунов А.Е., Постников Н.П., Чистяков А.Г. Механизмы граничного сканирования в неоднородных микропроцессорных системах // ChipNews. 2000. № 10.
2. IEEE Std 1149.1-2001. The Institute of Electrical and Electronics Engineers, Inc.-2001.
3. IEEE Std 1149.1(JTAG) Testability. Texas Instruments. SG.1997
4. IEEE 1149.1 Boundary-Scan Testing in Altera Devices. Altera Corporation, 2005. <www.altera.com>.
5. Tutorial covering Xilinx designed flows. <www/support.xilinx.com>
6. Computerworld, 14 ноября 2006.
7. Maurizio Rebaudengo. Boundary Scan Testing. COREP// <http://www.bolton.ac.uk/mind/corep/bst/bst-intro.html-2001>.

**ФОРМАЛЬНОЕ МОДЕЛИРОВАНИЕ «ОБЩИХ КРИТЕРИЕВ»
КАК ИНСТРУМЕНТ ВНЕДРЕНИЯ И ПРОДВИЖЕНИЯ
СТАНДАРТА ГОСТ Р ИСО/МЭК 15408****О.Е.Зайцев, А.В.Любимов**

В работе представлен обзор актуальной информации в области оценки защищенности информационных технологий, на основе которого обосновывается актуальность использования методов формального моделирования для внедрения и продвижения стандарта ГОСТ Р ИСО/МЭК 15408 «Общие Критерии» (ОК) в РФ. Дается постановка задачи формального моделирования ОК, проводится сравнительный анализ методов и обосновывается выбор методики и нотации функционального моделирования.

Введение

Национальный стандарт безопасности ГОСТ Р ИСО/МЭК 15408-2002 «Информационная технология. Методы обеспечения безопасности. Критерии оценки безопасности информационных технологий» (краткое название – «Общие критерии») начал действовать в России с 1 января 2004 года. В его основе лежит стандарт в области оценки безопасности информационных технологий (ИТ) «Common Criteria» (CC) 1999 г., разработанный под эгидой Международной организации по стандартизации. Исторически сложившимся названием в России этого стандарта является Руководящий документ «Безопасность информационных технологий» (РД БИТ). Семилетний опыт использования «Общих критериев» в мире и небольшой опыт, полученный при апробации в России, говорит о том, что применение методологии ОК способствует существенному повышению качества оценки и разработки продуктов и систем ИТ.

Объемы работ по оценке и сертификации ИТ, выполняемые в настоящее время за рубежом и планируемые в России, приводят к необходимости использования инструментальных программных средств поддержки деятельности по подготовке и проведению оценок. Современные методы разработки подобных средств предполагают широкое применение формальных моделей предметной области, по крайней мере, на стадиях специфицирования и высокоуровневого проектирования.

Решению перечисленных задач, как и многих других, а также внедрению и продвижению ОК в России способствует представление «Общих критериев» в виде множества разноплановых проекций. Такими проекциями являются три модели: функциональная модель деятельности по оценке защищенности ИТ, структурная модель защищенности ИТ и математическая модель компонентов защищенности ИТ. Как правило, функциональная модель является основной и наиболее применяемой на практике, поэтому здесь будет рассматриваться вопрос, касающийся построения функциональной модели защищенности ИТ, остальные формальные модели ОК будут рассмотрены в дальнейшем.

В настоящей работе обосновывается актуальность использования методов формального моделирования для внедрения и продвижения стандарта ГОСТ Р ИСО/МЭК 15408 в РФ, дается постановка задачи формального моделирования ОК, проводится сравнительный анализ методов и обосновывается выбор методики и нотации функционального моделирования.

Постановка задачи функционального моделирования ОК

Основной причиной создания международного стандарта ИСО/МЭК 15408 «Критерии оценки безопасности информационных технологий» (Common Criteria) являлась необходимость унификации и взаимного признания национальных стандартов в области безопасности информационных технологий. Кроме того, единый набор международных стандартов, разработанный для достижения этой цели, позволяет упростить принятие решений при покупке программных продуктов и предоставляет возможность корпоративным заказчикам приобрести системы с более надежной защитой [1].

ОК представляют собой базовый стандарт, содержащий методологию задания требований и оценки безопасности ИТ, а также систематизированный каталог требований безопасности. В качестве функциональных стандартов, в которых формулируются требования к безопасности определенных типов продуктов и систем ИТ, предусматривается использование профилей защиты (ПЗ), создаваемых на основе каталога требований. В ПЗ могут быть включены и другие требования, необходимые для обеспечения безопасности конкретного типа продуктов или систем ИТ. Изложенная в ОК методология формирования требований и универсальный каталог требований безопасности (функциональных и доверия) позволяют формировать наборы требований (ПЗ, задания по безопасности и пакеты) для различных типов продуктов и систем ИТ. Несмотря на то, что ОК направлены на оценку продуктов и систем ИТ и в документе содержится методология оценки, они включают не только методологию формирования требований для оценки, но и методологию формирования требований к разработчику по организации процесса разработки, по предоставлению материалов, необходимых для проведения оценки, а также другие обязанности разработчика в процессе проведения оценки [2]. Таким образом, применение ОК способствует существенному повышению качества оценки и разработки продуктов и систем ИТ.

Внедрение ОК в России спланировано поэтапно. До 2007 г. организации, в информационных системах которых циркулирует конфиденциальная информация, могли самостоятельно выбирать, по каким стандартам (старым или новым) проводить аттестацию, в отличие от организаций, которые обрабатывают в своих автоматизированных системах данные, содержащие сведения, представляющие собой государственную тайну. Для госструктур, которые имеют дело с подобной информацией, остаются в силе прежние РД Гостехкомиссии и требования ФСБ и бывшего ФАПСИ. Поскольку внедрение нового ГОСТ является частью правительственной программы по вступлению России в ВТО, а, как известно, при вступлении в эту организацию в стране-претенденте должны быть унифицированы некоторые стандарты, в том числе и в области информационной безопасности, в ближайшем будущем становится неизбежной масштабная деятельность по оцениванию и сертификации продуктов ИТ по стандарту ОК [3]. По этой причине уже сейчас существует необходимость подготовки значительного числа специалистов по ОК как для центров оценки и сертификации, так и для фирм-разработчиков, фирм-поставщиков и организаций-пользователей.

По прошествии некоторого времени использования стандарта ОК в области сертификации и оценки безопасности продуктов ИТ стало ясно, что предусмотренные стандартом средства не являются достаточно полными для полноценного их использования на практике и, безусловно, нуждаются в существенном расширении. Методология ОК в стандарте не имеет явного описания, ее элементы рассредоточены по тексту, который, вместе с сопутствующей нормативно-методической документацией, составляет около двух тысяч страниц. При этом значительная часть русскоязычной методической документации находится в стадии разработки или причислена к know-how и потому является недоступной. Такое положение дел является причиной недоступности методологии ОК для широкого круга пользователей. Вскоре необходимость в более полном описании и расширении средств регламентации деятельности по оценке была восполнена новым нормативным документом «Общая методология оценки» (ОМО) [4] – переводом английского варианта «Common Methodology for In-

formation Technology Security Evaluation» [5], сопровождающим ОК. Главным мотивирующим моментом создания ОМО стала необходимость унификации способов и приемов проведения оценки по ОК с целью выработки механизмов взаимного признания оценок.

Как известно основным предметом ОК являются элементы защиты (функции защищенности, свидетельства оценки, компоненты доверия к ним), в то время как основным предметом рассмотрения ОМО являются именно действия по оценке защищенности с использованием критериев и свидетельств оценки, определенных в ОК. В первую очередь ОМО предназначен для оценщиков, которые используют ОК, а также экспертов органов по сертификации, подтверждающих действия оценщиков. Также ОМО полезен заявителям на проведение оценки в целях получения исходной информации, разработчикам продуктов и систем ИТ, профилей защиты и заданий по безопасности, а также другим сторонам, заинтересованным в обеспечении безопасности ИТ [6].

Для реализации стандартов ОК и ОМО разработчикам пришлось рассмотреть общие подходы, методы и функции обеспечения защиты информации в организациях, а также описать, каким образом функции системы информационной безопасности обеспечивают выполнение требований конфиденциальности, целостности, достоверности и доступности информации [7]. Наличие таких обобщений позволяет применять методы научного анализа, в первую очередь – построение формальных моделей. В данной работе идет речь об одной из таких моделей – функциональной модели деятельности по оценке защищенности ИТ в рамках стандартов ОК и ОМО. Функциональная модель ОК должна отображать основные действия оценщика и разработчика, описанные, в основном, в ОМО. Актуальность задачи функционального моделирования методологии ОК обуславливают перечисленные ниже факторы.

Автоматизация многих действий по оценке и сертификации продуктов и систем ИТ возможна благодаря строгой регламентации деятельности оценщика и разработчика, описанной в основном в ОМО и частично в ОК. Для проектирования, разработки и сопровождения соответствующего программного обеспечения необходимо иметь функциональные спецификации в стандартизированной электронной форме. Также стандартизованная функциональная модель предоставляет удобные средства контроля версий стандарта и обеспечивает возможность прослеживания последствий принимаемых в новых версиях изменений вплоть до уровня конкретных операций, что существенно облегчает как работу сотрудников испытательных лабораторий при проведении оценки, так и работу разработчиков при подготовке к ней. Наименее подготовленной к предстоящему внедрению ОК группой пользователей являются заказчики ИС и покупатели готовых продуктов. Для них формализованное графическое представление ОК является кратким справочником.

Таким образом, представление базовых концепций ОК и их взаимосвязей в виде формальной структурированной функциональной модели является совершенно необходимым условием для продвижения и эффективного применения стандарта «Общие критерии».

Обзор зарубежных источников выявил лишь одну попытку функционального моделирования фрагментов ОК [8]. Однако при построении диаграмм реально не использовалась какая-либо определенная методика или метод, фактически они представляют собой иллюстрации, а не формализованную модель. Модель построена не по стандарту, что не позволяет говорить об ее дальнейшей применимости в области оценки и сертификации ИТ вследствие своей неполноты и субъективности. В отечественной литературе идея использования методики SADT в совокупности с методом DFD для построения формальной модели процессов оценки безопасности ИТ по стандарту ОК была предложена в [9]. Настоящая работа представляет собой ее обоснование.

Выбор и обоснование выбора метода функционального моделирования ОК

Проблема выбора методики функционального представления стандарта ОК при решении данной задачи не является критической, поскольку классическая методика SADT в данной области моделирования практически не имеет альтернативы. Получившая распространение в последние пять лет методика ARIS (Architecture of Integrated Information System) [10] ориентирована на процессы, протекающие в рамках сложной организационной структуры, а появившаяся практически одновременно методика CALS (Continuous Acquisition and Life cycle Support) [11] нацелена на представление сложных технологических цепочек процессов в многоагентной среде. Обе эти методики в поставленной задаче являются избыточными.

Методика моделирования SADT была разработана Дугласом Россом. Она представляет собой совокупность методов, правил и процедур, предназначенных для построения функциональной модели объекта какой-либо предметной области. Следует учитывать, что под методикой моделирования (как функционального, так и информационного) принято понимать «триаду»: 1) метод моделирования (способ анализа); 2) нотация моделирования (способ графического отображения результата анализа, т.е. собственно модели); 3) программное средство, поддерживающее первые два элемента триады. Однако обычно для краткости вместо термина «метод» используют термин «модель», подчеркивая тем самым основной результат анализа. Функциональная модель SADT отображает функциональную структуру объекта, т.е. производимые им действия и связи между этими действиями. Методика SADT основывается на следующих принципах:

- (1) графическое представление блочного моделирования. Графика блоков и дуг SADT-диаграммы отображает функцию в виде блока, а интерфейсы входа/выхода представляются дугами, входящими в блок и выходящими из него. Взаимодействие блоков друг с другом отображается посредством интерфейсных дуг, которые определяют, когда и каким образом функции выполняются и управляются;
- (2) строгость и точность выполнения правил SADT:
 - (а) ограничение количества блоков на каждом уровне декомпозиции (3–6 блоков);
 - (б) связность диаграмм (нумерация блоков);
 - (в) уникальность имен;
 - (г) синтаксические правила для графических примитивов (блоков и дуг);
 - (д) разделение входов и управлений (правило определения роли данных);
 - (е) отделение организации от функции, т.е. исключение влияния организационной структуры на функциональную модель [12].

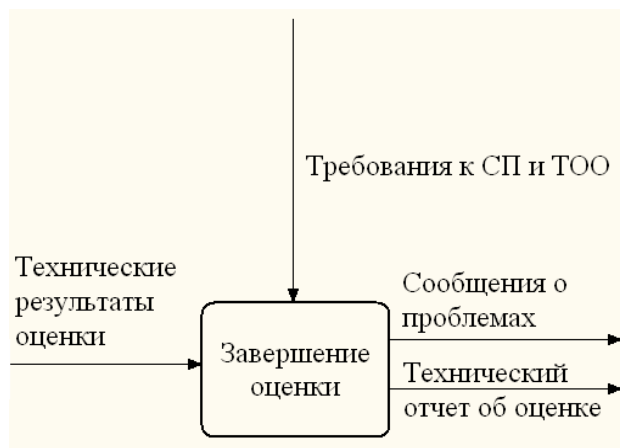


Рис. 1. Функциональный блок завершения оценки с интерфейсными дугами

Функциональная модель состоит из диаграмм, фрагментов текстов и глоссария, имеющих ссылки друг на друга. Диаграммы – главные компоненты модели отображения информационной системы (ИС), все функции ИС и интерфейсы на них представлены как функциональные блоки и дуги. Место соединения дуги с блоком определяет тип интерфейса. Управляющая информация входит в блок сверху, информация, которая подвергается обработке, входит в блок с левой стороны, а результаты выхода выходят с правой стороны блока. Таким образом, функциональный блок диаграммы «Процесс оценки безопасности продукта или системы ИТ» функциональной модели ОК будет выглядеть так, как показано на рис. 1.

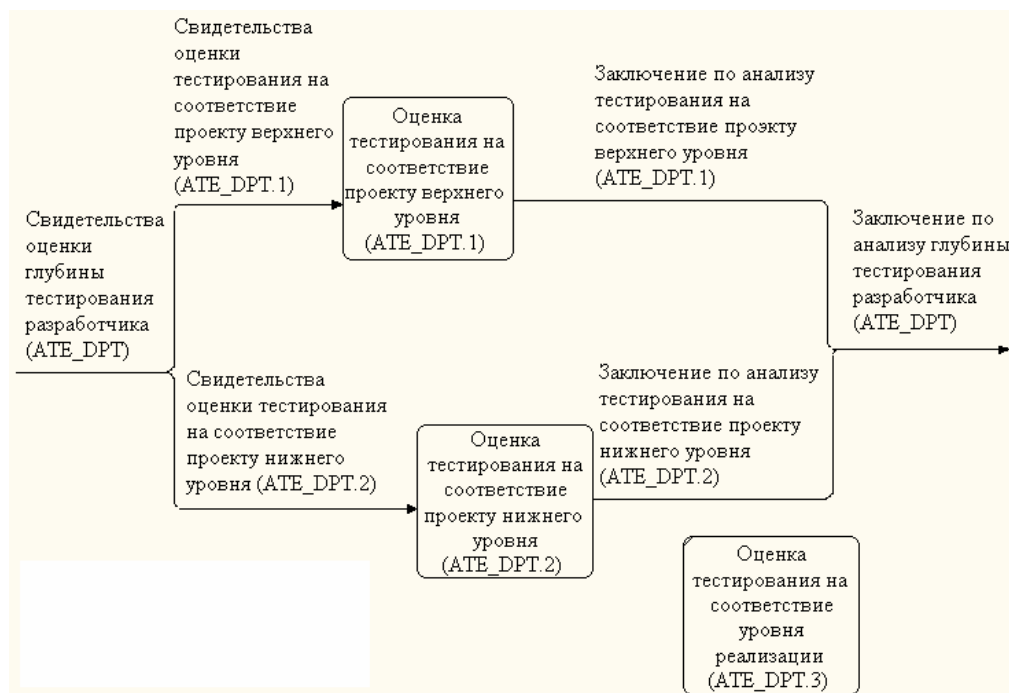


Рис. 2. Диаграмма «Оценка глубины тестирования разработчика (ATE_DPT)»

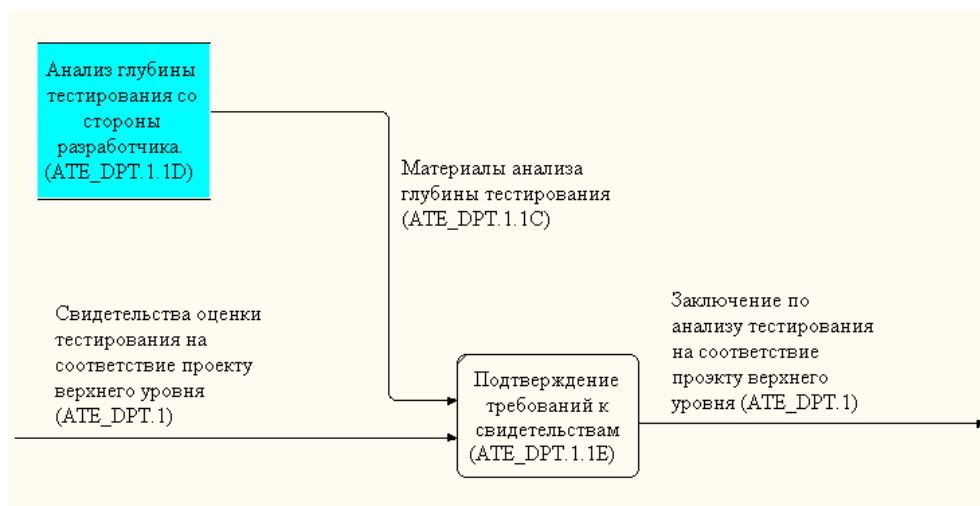


Рис. 3. Диаграмма «Оценка тестирования на соответствие проекту верхнего уровня (ATE_DPT.1)»

Важной особенностью методики SADT является постепенное увеличение уровней детализации по мере создания диаграмм. Каждый компонент модели может быть декомпозирован (детализирован) на другой диаграмме. Каждая диаграмма отображает строение блока на диаграмме предыдущего, более высокого уровня. На рис. 3 приведен

пример детализации блока АТЕ_DPT.1, изображенного на диаграмме АТЕ_DPT (рис. 2). Диаграмма АТЕ_DPT.1 является результатом детализации диаграммы АТЕ_DPT, что отображено в названии блоков.

Методика SADT предоставляет 3 метода моделирования: IDEF0 (Integration Definition for Function Modeling), IDEF3 и DFD. Анализируя и сравнивая между собой данные методы для решения поставленной задачи, однозначно выбирается метод DFD.

Действительно, метод IDEF0 моделирования бизнес-процессов фокусируется на высокоуровневом представлении операций и на факторах, которые контролируют эти операции. В рассматриваемой задаче все контролирующие факторы очевидны, а операции, наоборот, должны представляться на весьма низком уровне – вплоть до элементов или даже шагов действий оценщика. Поэтому метод IDEF0 не подходит. Такой же вывод, но по другим причинам, можно сделать и в отношении метода IDEF3, который фокусируется на бизнес-логике выполнения операций и на принятии решений, относящихся к выполнению определенной операции, а также на том, каким образом осуществляется синхронизация процессов. В поставленной задаче данной работы эти вопросы не имеют существенного значения, в первую очередь благодаря тем широким возможностям распараллеливания работ на независимые действия, которые обеспечиваются в рамках методологии ОК. К тому же метод IDEF3 практически лишен средств описания данных (документов, свидетельств, результатов действий), которые являются важнейшими компонентами процесса оценивания.

Метод DFD является модификацией метода IDEF0. Главной модификацией стала трактовка функциональных блоков (activities) как модулей программной системы. В остальном основная суть метода IDEF0, последовательная иерархическая декомпозиция, не изменилась, так как прекрасно вписывалась в область нисходящего проектирования программных систем. Впоследствии идея ICOM в отношении дуг оказалась излишней (на примере ИС): управления, ограничивающие или предписывающие условия выполнения функций модуля стали очевидными – это просто его программа; механизмы, показывающие, кто и с помощью чего выполняет функцию модуля – это компьютер и операционная система. Входы и выходы модулей стали равноправны – это потоки данных, передаваемых от модуля к модулю. Таким образом, все синтаксические ограничения были отброшены. Тем не менее, при такой трактовке нотация IDEF0 оказалась явно бедна – в ней не хватало нескольких элементов, которые и были добавлены.

Внешние источники (External References), располагаемые на контекстной диаграмме, позволяют явно описать внешние источники и приемники данных модели – внешние программные системы, внешние базы данных, внешние организации, операторы и т.д. Важно также, что можно использовать несколько экземпляров одного и того же внешнего источника на разных диаграммах модели. Это упрощает ее читаемость за счет существенного уменьшения числа потоков данных.

Хранилища данных (Data Stores), которые также можно использовать в нескольких экземплярах на разных диаграммах модели, позволяют описать «данные в покое». По существу, это прообразы таблиц базы данных проектируемой программной системы. Дуги, соединяющие их с модулями программной системы, моделируют информационный обмен приложений с базой данных.

Ссылки на другие страницы позволяют описывать передачу данных от модуля к модулю, избегая их «протаскивания» через все диаграммы верхнего уровня.

Таким образом, DFD-модели гораздо более информативны и более легко читаемы. Даже на контекстном уровне большая информативность DFD-диаграмм очевидна. В настоящее время диаграммы верхнего уровня иерархии, описывающие контекст деятельности организации в целом, выполняются в нотации IDEF0, а конкретные процессы, предназначенные для автоматизации на более низких уровнях иерархии, выполняются в нотации DFD. Таким образом, наилучшим выбором в задаче функционального

моделирования ОК является метод DFD, который фокусируется именно на операциях обработки данных – данных, необходимых для проведения операций или создаваемых какими-либо операциями; данных для групп или организаций, которые либо предоставляют, либо получают данные; потоком данных между различными действиями, видами или подвидами действий или операциями, а также между операциями и вовлеченными хранилищами данных. Метод DFD, по сравнению с IDEF0, позволяет, во-первых, гораздо более полно отразить на диаграммах роли сущностей, которые иницируют, выполняют, заканчивают или используют результаты выполнения процессов, а во-вторых, дает возможность гораздо более полно представить обмен ресурсами (в частности – документами) между сущностями процессов.

Таким образом, задача выбора методики моделирования деятельности по оценке защищенности ИТ по стандарту «Общие критерии» решена.

Научная новизна настоящей работы состоит в том, что описанная выше методика SADT и ее метод графического отображения DFD позволили построить полную функциональную модель оценки защищенности ИТ по стандарту «Общие критерии», детализировав эту модель до уровня представления элементов действий оценщика и разработчика. На основании идей и исходных материалов по функциональному моделированию ОК, предложенных в [9], а также внесенных коррективов в методику SADT, в связи с особенностями моделируемого объекта, была получена модель, в состав которой входит 91 функциональная диаграмма, 241 процесс в иерархии и 410 ресурсов.

Заключение

Национальный стандарт безопасности ГОСТ Р ИСО/МЭК 15408-2002 «Информационная технология. Методы обеспечения безопасности. Критерии оценки безопасности информационных технологий» был принят в России 1 января 2004 года. Семилетний опыт использования «Общих критериев» в мире и небольшой опыт, полученный при апробации в России, говорит о том, что применение методологии ОК способствует существенному повышению качества оценки и разработки продуктов и систем ИТ. Таким образом, оценка защищенности ИТ по ОК является перспективным направлением, а проблемы, связанные с использованием ОК, очень актуальными.

Для внедрения, продвижения и эффективного использования данного стандарта необходимо представить «Общие критерии» в виде формализованных, структурированных моделей, таких как структурная, функциональная и математическая модель. Также к представлению ОК в формальной форме приводят работы по оценке и сертификации ИТ, выполняемые в настоящее время за рубежом и планируемые в России. В данном случае необходимо использовать инструментальные программных средства поддержки деятельности по подготовке и проведению оценок, а современные методы разработки подобных средств предполагают широкое применение формальных моделей. Представление ОК в виде формальной модели позволяет решать ряд практических задач в области оценки защищенности продуктов и систем ИТ и сертификации.

В данной работе речь шла о функциональной модели, так как она является основной и наиболее используемой на практике. Для построения функциональной модели ОК изначально необходимо было выбрать метод моделирования и обосновать данный выбор, что и было сделано. Исходя из сравнительного анализа методик и методов моделирования, обоснованно была выбрана методика SADT и ее нотация DFD.

Литература

1. Рэдклифф Д. Одна мера безопасности на всех. // Computerworld. 2000. № 12. С.76–80.

2. ГОСТ Р ИСО/МЭК 15408-1-2002. Информационная технология. Методы и средства обеспечения безопасности. Критерии оценки безопасности информационных технологий. Часть 1. Введение и общая модель. Госстандарт России, Москва, 2002.
3. Просяников Р.Е., Савельев М.С. ГОСТ 15408: Станут ли общими «Общие критерии». – Secure Networks, 2005. <http://www.secure.com.ru/modules.cfm?block=portal_b_documents&id=A7ZE70A3316950558>
4. РД Безопасность информационных технологий. Общая методология оценки безопасности информационных технологий (проект). – ФСТЭК России, 2005.
5. [CEM] Common Methodology for Information Technology Security Evaluation. Evaluation Methodology. January 2004. Version 2.2. Revision 256. CCIMB-2004-01-004.
6. Кобзарь М., Сидак А. Методология оценки безопасности информационных технологий по общим критериям. – JetInfo, № 6 (133), 2004. сс.2-16.
7. Афанасьев В.Н. Общие критерии безопасности информационных систем. – Международная студенческая школа-семинар "Новые информационные технологии", г. Судак, 14 – 21 мая, 2003, Тезисы докладов XI в 2-х томах - М.: МГИЭМ, 2003. 641с.
8. Prieto-Diaz, R. The Common Criteria Evaluation Process. Process Explanation, Shortcomings, and Research Opportunities. - Commonwealth Information Security Center Technical Report CISC-TR-2002-03, December 2002 – CISC, James Madison University, USA. 62 p.
9. Любимов А.В. Функциональная структура общих критериев оценки безопасности информационных технологий. – Труды 9-й научно-технической конференции "Теория и технология программирования и защиты информации. Применение вычислительной техники." Санкт-Петербург, 18 мая 2005 г. сс.20–24.
10. Шеер Август-Вильгельм. Бизнес-процессы. Основные понятия. Теория. Методы. Весть - МетаТехнология, 1999. 152 с.
11. Кобзарь М., Сидак А. Стандартизация безопасности ИТ. Мир Связи. – № 3, 2003. сс.7-16.
12. Вендров А.М. CASE-технологии. Современные методы и средства проектирования информационных систем. – CIT Forum, 1997. – http://citforum.ru/go/iframe/citforum_left.html?/database/case/index.shtml

ОБЕСПЕЧЕНИЕ ИЗБЫТОЧНОСТИ ИНФОРМАЦИОННЫХ ПОЛЕЙ АДАПТИВНЫХ КЛАССИФИКАТОРОВ ДЛЯ СИСТЕМ ЗАЩИТЫ ИНФОРМАЦИИ

Ф.Г. Нестерук, Г.Ф. Нестерук, Л.Г. Осовецкий

Рассмотрены вопросы повышения избыточности информационных полей нейро-нечетких сетей, необходимой для реализации основных достоинств нейросетевой элементной базы, таких как адаптивность, функциональная устойчивость, распределенное избыточное хранение информации. Показаны пути введения избыточности в информационные поля нейро-нечетких сетей, используемых для обеспечения безопасности (ИБ) систем информационных технологий (ИТ).

Введение

Обратной стороной специализации слоев нечеткой нейронной сети (НС), обеспечивающей «прозрачность» информационного поля для последующего анализа, является отсутствие информационной избыточности, что негативно сказывается на функциональной устойчивости и защищенности информационных полей НС к деструктивным воздействиям.

При сохранении удобной для анализа специализации отдельных слоев нейро-нечетких сетей в соответствии с правилами нечеткого логического вывода необходимо ввести избыточность в информационное поле нейро-нечеткого классификатора [1]. Избыточность информационного поля создает предпосылки для распределенного хранения информации в структурированных полях нечеткой НС в виде системы комплементарных связей [2], а структурная специализация слоев нейро-нечеткой сети позволяет анализировать результаты обучения информационных полей НС.

Недостатком нейро-нечетких сетей является низкая избыточность информационных полей, что негативно сказывается на функциональной устойчивости средств защиты к дестабилизирующим воздействиям. Следует использовать подходы введения избыточности в информационные поля НС, связанные с комплементарностью представления и дублированием информации [3], а также использованием информации о состоянии системы информационной безопасности (СИБ) [4]. Рассмотрим пути введения избыточности в информационные поля нейро-нечетких классификаторов.

Комплементарность представления и дублирование информации при решении задачи классификации

Введение избыточности в информационные поля нейро-нечеткого классификатора базируется на механизмах обеспечения защищенности биосистемы [5]:

- представление информации структурированными полями,
- информационной избыточности вследствие комплементарного дублирования структурированных информационных полей,
- информационной избыточности вследствие повторяющихся фрагментов информационного поля.

Избыточность информационного поля обеспечивает возможность для распределенного хранения знаний в структурированных полях нейро-нечеткой сети, а специализация слоев НС позволяет анализировать результаты обучения информационных полей НС.

В качестве аппарата для формализации преобразований над нечеткими высказываниями можно использовать аналог нормальных форм, а именно дизъюнктивной (ДНФ) и конъюнктивной (КНФ) нормальных форм [3, 4]. При этом правила логического вывода If-Then, описывающие базу знаний экспертов безопасности, представляются в виде двух систем правил – конъюнктивной и дизъюнктивной, которые отражаются в

специализации формальных нейронов (ФН) в слое композиции нейро-нечеткого классификатора (рис. 1).

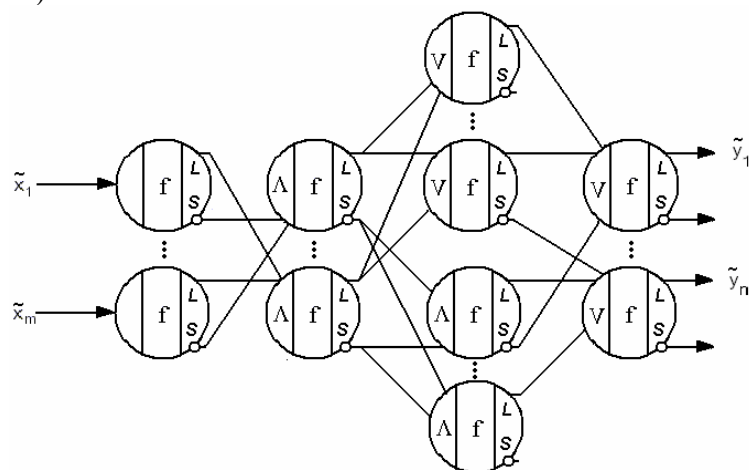


Рис. 1. Структура избыточного нейро-нечеткого классификатора

В первую группу включают нечеткие ФН «ИЛИ», реализующие операцию $\max$ над минтермами, во вторую группу – нечеткие ФН «И», реализующие операцию $\min$ над макстермами. В результате на выходах $L(\tilde{x}_i)$ и $S(\tilde{x}_i)$ первой группы из ФН «ИЛИ» будет реализовано, соответственно, прямое и инверсное представление системы нечетких правил, а на выходах $L(\tilde{x}_i)$ и $S(\tilde{x}_i)$ второй группы из ФН «И» - соответственно, инверсное и прямое представление системы нечетких классификационных правил. Выделение двух групп ФН в слое композиции позволяет объединить на нечетких ФН «ИЛИ» дополнительного 4-го слоя нейро-нечеткого классификатора одноименные правила реализуемой системы нечетких классификационных правил.

Введение избыточности в информационное поле нейро-нечеткого классификатора сохраняет «прозрачность» для анализа результатов нейросетевых средств защиты информации.

Использование информации о состоянии СИБ при решении задачи классификации

Повышение избыточности информационных полей нейро-нечетких сетей достигается также за счет использования сведений о состоянии системы безопасности в качестве одного из источников входных данных. Увеличение размерности входных данных обеспечивается добавлением к входному вектору X вектора Z текущего состояния СИБ (рис. 2).

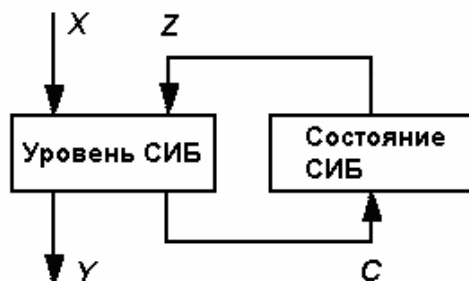


Рис. 2. Коррекция адаптивного уровня системы информационной безопасности

Подобная коррекция структуры системы защиты вызывает не только увеличение размерности входных данных классификатора, но и расширяет систему нечетких пра-

вил логического вывода, которая учитывает не только координаты входного вектора X , но и координаты вектора Z текущего состояния СИБ, что, в свою очередь, приводит к возрастанию избыточности информационного поля нейро-нечеткого классификатора. В процессе работы классификатора производится не только идентификация вектора Y по векторам X и Z , но и формируются предложения C по изменению состояния системы защиты компьютерной сети.

Квазилогические нейро-нечеткие классификаторы

Достоинство нейро-нечетких средств защиты информации обусловлено доступностью для анализа их информационных полей [6, 7]. Однако из процесса обучения исключается большинство связей информационного поля нейро-нечеткой сети, что ставит под сомнение распределенный характер представления знаний в информационном поле нейро-нечеткого классификатора сети. Предложено [8] использовать арифметические ФН для реализации операций $\max$ и $\min$ над взвешенными входными сигналами, отчасти подобные логическим операциям дизъюнкции и конъюнкции.

Квазилогические формальные нейроны – это арифметические ФН, во входных цепях которых реализованы дополнительные операции:

- сортировка взвешенных входных значений,
- последующее масштабирование значений посредством коэффициентов значимости.

Квазилогические формальные нейроны MIN эмулируют логическую операцию конъюнкции путем сортировки взвешенных входных значений ФН в порядке возрастания, масштабирования взвешенных входных значений путем умножения на монотонно возрастающий ряд коэффициентов значимости (рис. 3).

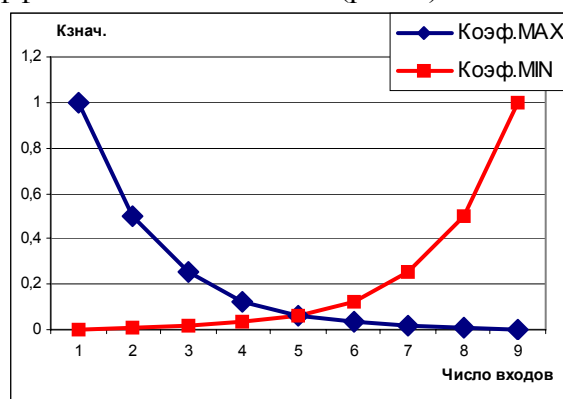


Рис. 3. Степенная зависимость Кзнач

Полученные значения суммируются и затем сжимаются функцией активации. То есть в квазилогических ФН MIN (рис. 4) меньшее взвешенное значение умножается на большее значение коэффициента важности.

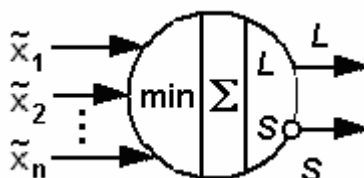


Рис. 4. Квазилогический нейрон MIN

Квазилогические формальные нейроны MAX эмулируют логическую операцию дизъюнкции путем сортировки взвешенных входных значений ФН в порядке возрастания, масштабирования взвешенных входных значений умножением на монотонно убывающий ряд коэффициентов значимости.

вающий ряд коэффициентов значимости (рис. 3). Отмасштабированные значения суммируются и затем сжимаются функцией активации. То есть в квазилогических ФН МАХ (рис. 5) большее взвешенное значение умножается на большее значение коэффициента важности.

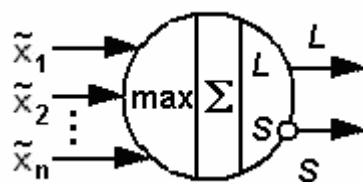


Рис. 5. Квазилогический нейрон МАХ

Квазилогические ФН позволяют при сохранении специфики операций конъюнкции и дизъюнкции реализовать распределенное хранение знаний специалистов безопасности в информационных полях нейро-нечетких средств защиты.

Если применить подход комплементарного дублирования для увеличения избыточности информационных полей НС к квазилогическим нейро-нечетким классификаторам, то получим аналогичную рис. 1 структуру НС на базе квазилогических ФН (рис. 6).

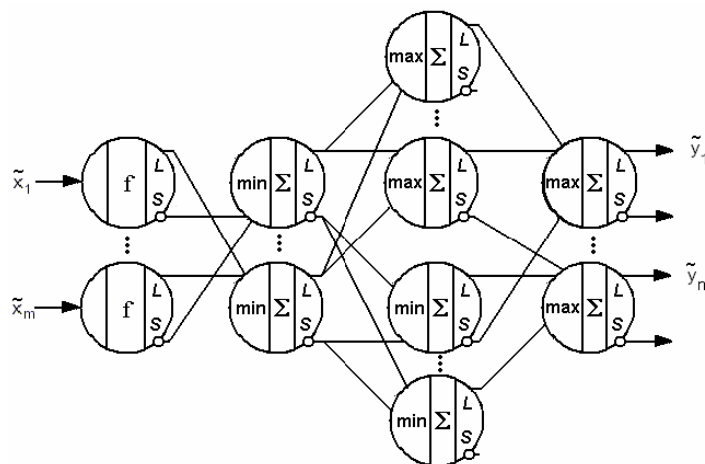


Рис. 6. Структура избыточного нейро-нечеткого классификатора

Для моделирования процессов обучения нейро-нечеткого классификатора используем несложную классификационную задачу, определяющую совпадение нечетких координат x_1 и x_2 входного вектора (логическая функция *Равнозначность*). Система конъюнктивных правил логического вывода будет иметь вид

R_1 : if $\tilde{x}_1$ is S and $\tilde{x}_2$ is S , then $\tilde{y} = L$,

R_2 : if $\tilde{x}_1$ is S and $\tilde{x}_2$ is L , then $\tilde{y} = S$,

R_3 : if $\tilde{x}_1$ is L and $\tilde{x}_2$ is S , then $\tilde{y} = S$,

R_4 : if $\tilde{x}_1$ is L and $\tilde{x}_2$ is L , then $\tilde{y} = L$.

Для формирования в квазилогическом нейро-нечетком классификаторе информационного поля, соответствующего реализуемой функции, проведено обучение НС по методу обратного распространения ошибки. Рис. 7 иллюстрирует динамику снижения ошибки при обучении квазилогической НС, в которой отсутствуют инверсные связи (с выходов S), а рис. 8 – процесс обучения квазилогической НС (рис. 6), в которой имеются как прямые, так и инверсные связи.

Как показало компьютерное моделирование, проведенное группой студентов кафедры БИТ в составе Доскача В.О., Инюшина И.М., Леонтьева Р.В., втором случае

нейронная сеть на базе квазилогических ФН обучается значительно быстрее, чем НС только с прямыми связями. Как известно, оперативность обучения и возможность последующего анализа информационного поля НС являются первостепенными атрибутами адаптивных средств защиты информации.

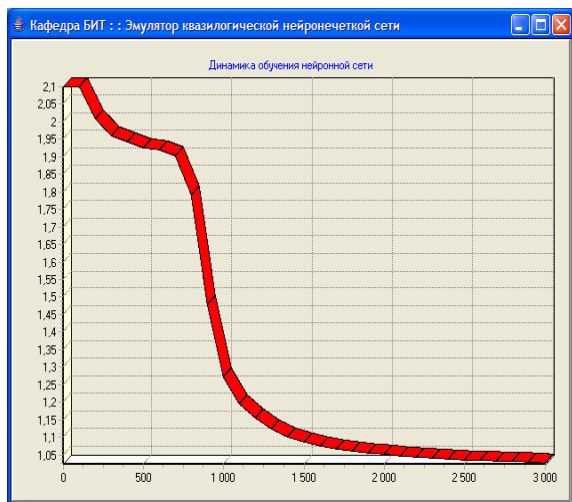


Рис.7. Динамика обучения НС с прямыми связями

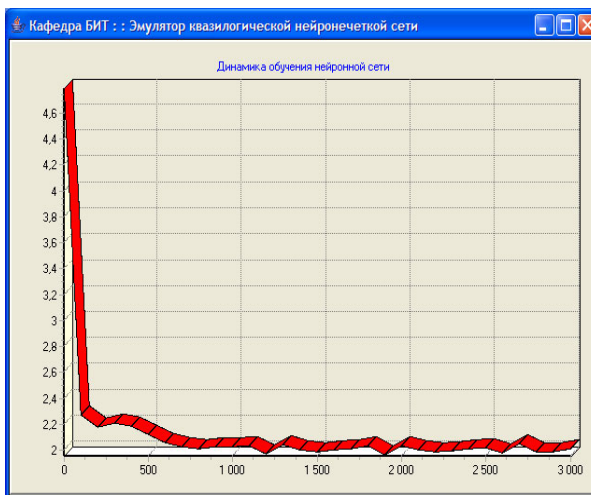


Рис.8. Динамика обучения НС с инверсными связями

Литература

1. Нестерук Г.Ф., Куприянов М.С., Елизаров С.И. К решению задачи нейро-нечеткой классификации // Сб. докл. VI Междунар. конф. SCM'2003. – СПб: СПбЭТУ, 2003. Т. 1. С. 244–246.
2. Нестерук Ф.Г., Осовецкий Л.Г., Нестерук Г.Ф., Воскресенский С.И. К моделированию адаптивной системы информационной безопасности // Перспективные информационные технологии и интеллектуальные системы. 2004. №4. С.25 –31.
3. Нестерук Г.Ф., Осовецкий Л.Г., Нестерук Ф.Г., Воскресенский С.И., Грибачев В.П. Информационная избыточность нейро-нечетких средств обеспечения безопасности // Вопросы защиты информации. 2005. № 3. С. 12–16.
4. Нестерук Г.Ф., Молдовян А.А., Нестерук Ф.Г., Воскресенский С.И., Костин А.А. Повышение избыточности информационных полей адаптивных классификаторов системы информационной безопасности // Специальная техника. 2006. № 1. С. 60–63.
5. Лобашев М. Е. Генетика. – Л.: Изд-во ЛГУ, 1969.
6. Круглов В.В., Борисов В.В. Искусственные нейронные сети. Теория и практика. / 2-е изд. – М.: Горячая линия - Телеком, 2002.
7. Fuller R. Neural Fuzzy Systems. - Abo: Abo Akademi University, 1995.
8. Нестерук Л.Г., Нестерук Г.Ф., Молдовян А.А., Нестерук Ф.Г. Анализ нейросетевой элементной базы для построения систем защиты информации // Вопросы защиты информации. 2006 (в печати).

МОДЕЛИ ПРОЦЕССА ОБНАРУЖЕНИЯ УГРОЗ ПРИ ДОСТУПЕ К РЕСУРСАМ КЛАСТЕРНОЙ СИСТЕМЫ

А.В. Птицын

Рассматриваются типовые модели процесса обнаружения угроз при доступе к ресурсам кластерной системы, учитывающие различные комбинации комплексирования механизмов защиты, а также иллюстрируются принципы и приемы построения моделей функционирования систем защиты информации кластеров.

Введение

Постоянное развитие архитектуры кластерных систем и действующих стандартов в области защиты информации требует уточнения принципов построения систем защиты информации кластеров и выявления необходимого набора механизмов защиты информации в кластерных системах [1, 2]. Для выявления наиболее эффективных вариантов построения систем защиты информации кластеров необходимо построение математических моделей [3].

За концептуальную основу построения систем защиты информации кластеров следует принять представление защищаемых объектов в поле угроз как высокопроизводительных компьютеров, так и распределенных систем.

Рассмотрим базовый набор моделей функционирования систем защиты информации кластеров, учитывающих их технологические особенности.

Построение моделей процесса обнаружения угроз при доступе к ресурсам кластерной системы

Представленные модели описывают процесс функционирования системы защиты информации кластеров при обнаружении угроз на этапе доступа к ресурсам кластера. Предполагается, что доступ к системе уже получен, и теперь угроза направлена на ресурсы кластера, например, осуществляется попытка запуска посторонней задачи или чтения файла. Кластер здесь уже рассматривается как вычислительная система. К ресурсам кластера относится весь кластер целиком (может быть разрешен или запрещен запуск задач на кластере, администрирование кластера, его конфигурирование), узлы кластера (узел кластера представляет собой отдельный компьютер, к которому можно получить доступ, осуществить его администрирование и конфигурацию, запустить какие-либо программы), локальные ресурсы узлов кластера (файловые системы, программное и аппаратное обеспечение).

Наличие возможности доступа к тому или иному ресурсу кластера определяется, во-первых, наличием данного ресурса, во-вторых, принципом построения и организации работы кластера. Поэтому те или иные механизмы защиты могут просто не иметь смысла в определенных реализациях кластерных систем. Например, узлы кластера могут не иметь жестких дисков, может быть вообще не предусмотрен непосредственный доступ к узлу кластера и его локальным ресурсам, а только косвенный – через запуск задачи для решения на кластере и распределение вычислительной работы механизмами кластера.

При проверке прав доступа к тому или иному ресурсу могут применяться самые разные модели управления доступом: дискреционная модель, мандатная модель, модель управления доступом на основе ролевой политики, модели на основе списков санкционированных событий и другие. Все они имеют свою специфику, что может учитываться в процессе детализации базовых моделей. Отметим только, что определенные права доступа могут быть привязаны к самому объекту доступа, например, права на файл записываются в атрибутах самого файла, и для их проверки необходимо об-

ратиться к самому объекту, т.е. к узлу, на котором расположен файл. Некоторые права и полномочия хранятся независимо от объектов доступа, например, информация о праве изменять системное время узла может храниться как локально, так и на другой машине, где указано, что данный пользователь является администратором системы.

Рассмотрим базовые модели функционирования системы защиты информации при попытке получения доступа к различным ресурсам. Приведем список введенных обозначений:

$f^u_{лпп}(k_{лпп}), k_{лпп} = 1, \dots, K^H_{лпп}$ – плотность распределения дискретного времени выполнения механизма контроля доступа с локальной проверкой прав в том случае, когда угроза не обнаружена;

$f_{лпп}(k_{лпп}), k_{лпп} = 1, \dots, K_{лпп}$ – плотность распределения дискретного времени выполнения механизма контроля доступа с локальной проверкой прав при обнаружении угрозы;

$P_{лпп}$ – вероятность обнаружения угрозы при контроле доступа с локальной проверкой прав;

$f_{св}(k_{св}), k_{св} = 1, \dots, K_{св}$ – плотность распределения дискретного времени, затраченного на сетевое взаимодействие (обращение к удаленной базе прав доступа);

$f^u_{рег}(k_{рег}), k_{рег} = 1, \dots, K^H_{рег}$ – плотность распределения дискретного времени выполнения механизма регистрации событий в том случае, когда угроза не обнаружена;

$f_{рег}(k_{рег}), k_{рег} = 1, \dots, K_{рег}$ – плотность распределения дискретного времени выполнения механизма регистрации событий при обнаружении угрозы;

$f^u_{спп}(k_{спп}), k_{спп} = 1, \dots, K^H_{спп}$ – плотность распределения дискретного времени выполнения механизма контроля доступа с проверкой прав по сети в том случае, когда угроза не обнаружена;

$f_{спп}(k_{спп}), k_{спп} = 1, \dots, K_{спп}$ – плотность распределения дискретного времени выполнения механизма контроля доступа с проверкой прав по сети при обнаружении угрозы;

$P_{спп}$ – вероятность обнаружения угрозы при контроле доступа с проверкой прав по сети;

$f^u_{авт}(k_{авт}), k_{авт} = 1, \dots, K^H_{авт}$ – плотность распределения дискретного времени выполнения механизма авторизации в том случае, когда угроза не обнаружена;

$f_{авт}(k_{авт}), k_{авт} = 1, \dots, K_{авт}$ – плотность распределения дискретного времени выполнения механизма авторизации при обнаружении угрозы;

$P_{авт}$ – вероятность обнаружения угрозы при авторизации;

$f^u_{пнр}(k_{пнр}), k_{пнр} = 1, \dots, K^H_{пнр}$ – плотность распределения дискретного времени выполнения механизма проверки наличия ресурса в том случае, когда угроза не обнаружена;

$f_{пнр}(k_{пнр}), k_{пнр} = 1, \dots, K_{пнр}$ – плотность распределения дискретного времени выполнения механизма проверки наличия ресурса при обнаружении угрозы;

$P_{пнр}$ – вероятность обнаружения угрозы при проверке наличия ресурса;

$f^u_{шф}(k_{шф}), k_{шф} = 1, \dots, K^H_{шф}$ – плотность распределения дискретного времени выполнения механизма шифрования в том случае, когда угроза не обнаружена;

$f_{шф}(k_{шф}), k_{шф} = 1, \dots, K_{шф}$ – плотность распределения дискретного времени выполнения механизма шифрования при обнаружении угрозы;

$P_{шф}$ – вероятность обнаружения угрозы при шифровании.

На рис. 1 представлена модель, отражающая простейший вариант работы системы защиты информации при доступе к ресурсу. В данном случае информация о правах

доступа к объекту хранится локально, скорее всего, она привязана к объекту. Примером может служить доступ к файлу или некоторому устройству. Сначала осуществляется доступ к объекту по сети (если осуществляется доступ к локальному объекту, то данная процедура отсутствует). Далее происходит проверка прав доступа на основе информации, хранимой локально относительно объекта доступа. При любом исходе проверки результат учитывается системой регистрации событий.

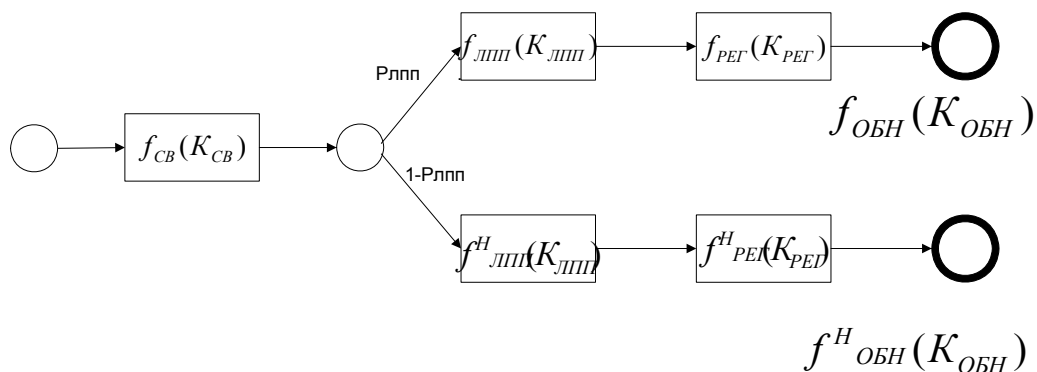


Рис. 1. Базовая модель процесса обнаружения угрозы при доступе к ресурсам кластерной системы

На рис. 2 представлена модель для варианта доступа к ресурсу, информация о правах на который может храниться как локально, так и на удаленной машине. При этом осуществляется доступ к ресурсу, проверяется локальная информация о правах данного пользователя на доступ к ресурсу. Уже на этом этапе в доступе пользователю может быть отказано без дальнейшей проверки – пользователю может быть указан явный запрет на использование данного ресурса. Если же пользователь сохраняет возможность доступа (хотя бы гипотетическую), осуществляется проверка информации о правах доступа, хранящихся на другой машине (например, сервере безопасности или контроллере домена), чтобы уточнить локальную информацию.

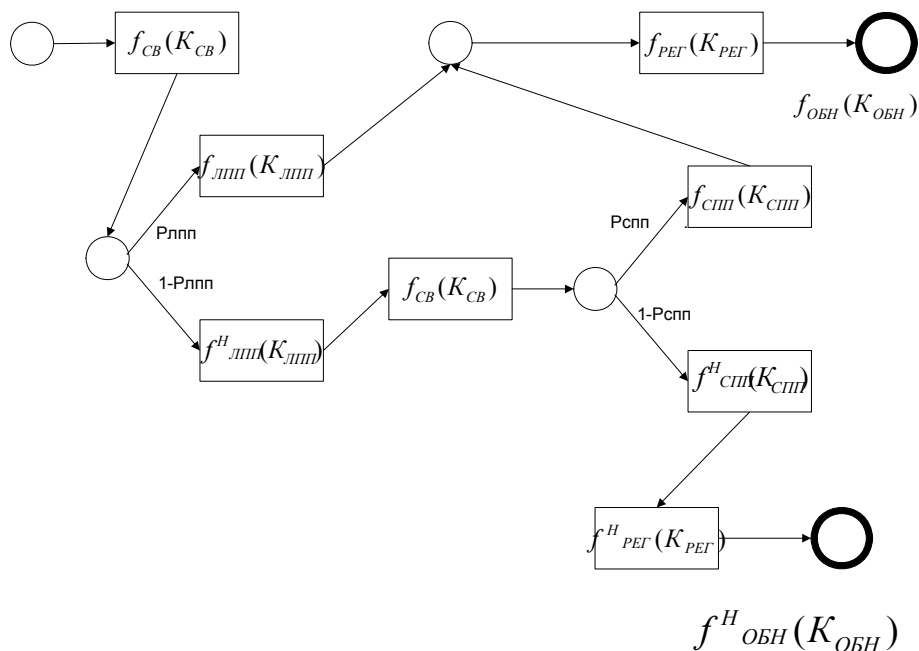


Рис. 2. Модель последовательного процесса обнаружения угрозы при доступе к ресурсам кластерной системы

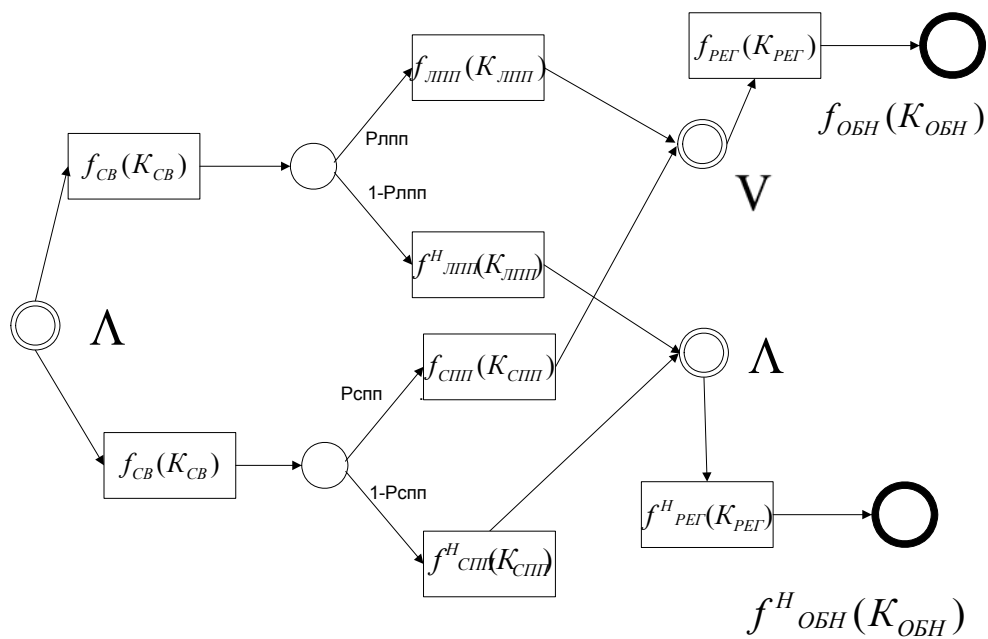


Рис. 3. Модель параллельного процесса обнаружения угрозы при доступе к ресурсам кластерной системы

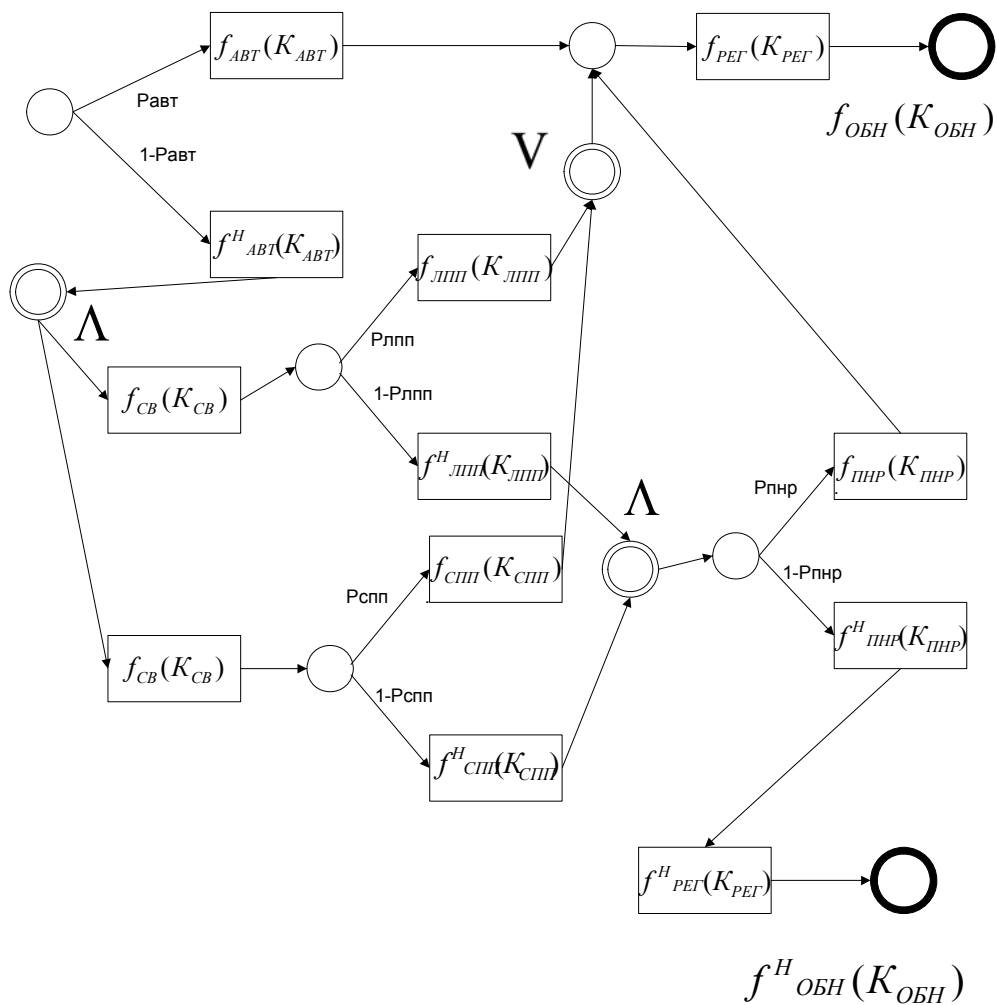


Рис. 4. Модель процесса обнаружения угрозы при доступе к ресурсам кластерной системы с авторизацией

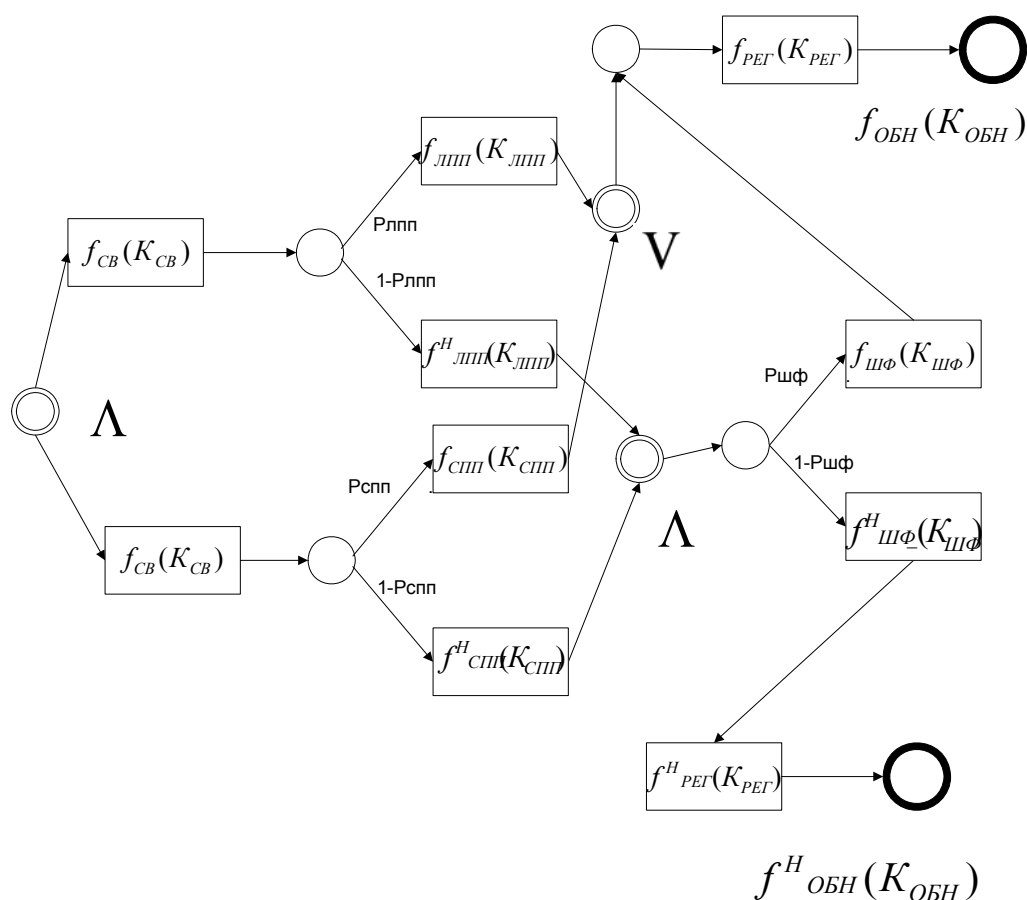


Рис. 5. Модель процесса обнаружения угрозы при доступе к ресурсам кластерной системы с шифрованием

На рис. 3 представлена такая же ситуация, но запрос к локальному и удаленному источникам информации о правах доступа осуществляется параллельно, что позволяет ускорить процесс предоставления доступа к объекту.

На рис. 4 изображена модель, содержащая следующие усложнения по сравнению с предыдущими – перед попыткой получения доступа к ресурсу пользователь проходит дополнительную авторизацию. Это может происходить, если пользователь сознательно воспользовался заниженными правами при доступе к системе, чтобы оградить себя от возможности случайной ошибки, или в случае принудительной вторичной авторизации при доступе к особенно важным ресурсам. Также здесь добавлен и механизм проверки наличия ресурса, т.е. его готовности. Важный ресурс может включать дополнительные ограничения на доступ к нему, например, запрещен одновременный доступ или запрещен доступ при определенной занятости ресурса.

На рис. 5 в модель добавлен новый механизм – механизм шифрования. Здесь подразумевается такой тип доступа к объекту, который предполагает обязательное шифрование информации. Примером может служить запрос на чтение файла с передачей его содержимого по сети – в этом случае для файлов, содержащих секретную информацию, осуществляется принудительная криптографическая обработка информации. Другим примером может служить запрос на чтение файла при использовании шифрующей файловой системы. В этом случае механизм шифрования будет осуществлять обратные действия – расшифровывать содержимое файла.

Заключение

Представленные модели не отражают всех возможных комбинаций комплексирования механизмов защиты, но выделяют типовые случаи и иллюстрируют принципы и приемы построения моделей функционирования систем защиты информации кластеров.

Полученные модели позволяют анализировать характеристики систем защиты информации кластеров и улучшать их на любом этапе жизненного цикла.

Литература

1. Корнеев В.В. Параллельные вычислительные системы. – СПб: Нолидж, 1999. – 320 с.
2. Домарев С. Безопасность информационных технологий. – СПб: Диасофт, 2002. – 240 с.
3. Головкин Б.А. Расчет характеристик и планирование параллельных вычислительных процессов – М.: Радио и связь, 1983. – 272 с.

СТРУКТУРНОЕ МОДЕЛИРОВАНИЕ КОНКУРЕНТНОЙ БОРЬБЫ СУБЪЕКТОВ КОРПОРАТИВНОЙ СИСТЕМЫ ЗА РЕСУРСЫ

М.В. Береговой, Л.Г. Осовецкий

В статье исследовано многосубъектовое взаимодействие между субъектами корпоративной системы с использованием элементов тактической и стратегической конкуренции. Предложены структурные модели конкурентной борьбы субъектов корпоративной системы за ресурсы, принципы возникновения конкуренции и языковые связи между субъектами.

Введение

Технический прогресс в области информационных технологий (ИТ) и телекоммуникационных систем (ТКС) привел к созданию нового объекта взаимодействия между субъектами корпоративной системы – многосубъектового взаимодействия с использованием элементов тактической и стратегической конкуренции. В настоящее время создалась ситуация, когда качество взаимодействия во многом зависит от качества решения тактических и стратегических вопросов конкурирующего взаимодействия в корпоративной среде. В процессе развития информационных технологий наличие конкурирующего взаимодействия между компонентами системы и борьба за установление контроля отдельных компонентов над однородными общесистемными ресурсами привело к обострению вопросов защиты и безопасности информации.

Целевое назначение систем защиты информации (СЗИ) как средства обеспечения накопления ресурсов, существования и наличия конкуренции – это основные факторы, которые позволяют построить модели корпоративной системы, возникновения конкуренции и элементов информационного взаимодействия, а также обосновать их необходимость и назначение.

В контексте сказанного видно, что конкуренция за обладанием ресурсами – один из основополагающих факторов развития корпоративной системы. Основываясь на корпоративной теории информации [1], в которой описаны методы построения корпоративных моделей информационной системы и моделях эффективного распределения ресурсов [2], авторами предложены модели конкурентной борьбы субъектов корпоративной системы за ресурсы, выполненные в форме структурных моделей [3]. Для моделирования были выбраны методика OOAD и метод OMG UML.

Большое значение в указанных моделях приобретает обоснование языков обмена информацией в корпорации, числа корпораций и языков, а также необходимого и достаточного уровня их защищенности.

Понятие корпорации, ресурсов и системы

Корпоративную систему можно рассматривать как совокупность субъектов, обладающих частью общих характеристик. Между такими субъектами существуют информационные взаимодействия, т.е. взаимный или односторонний обмен данными. Из этого утверждения следует, что корпоративную систему можно представить как информационную систему, обладающую совокупностью субъектов, осуществляющих информационное взаимодействие. Особенность корпоративной системы как информационной системы заключается в корпоративном характере информационных процессов.

Информационное взаимодействие возникает только при наличии других подобных и конкурирующих субъектов, т.е. системы субъектов, что является основой для организации совместной и конкурентной борьбы за ресурсы развития и существования как системы в целом, так и субъектов этой системы. Естественно, подразумевается, что субъекты системы находятся в постоянном информационном взаимодействии между собой.

Субъекты такой системы разделяются по уровням сложности, и используемые ими ресурсы также можно разделить по уровням комплексности, т.е. чем сложнее субъект, тем большее разнообразие ресурсов он может использовать.

Объединение во временные объединения – корпорации – связано с необходимостью обеспечения «секретности» по Шеннону от других членов [4]. Поэтому свойства языка корпорации определяются числом возможных объединений в корпорации при соблюдении определённых правил безопасности. Также имеет смысл учитывать как возможность субъектов объединяться в корпорацию, так и отсутствие таковой. Здесь имеется в виду территориальная разрозненность субъектов и параметры каналов связи. Логично, что подобное объединение при сильном удалении субъектов друг от друга потребует значительных затрат на физическое обеспечение взаимодействия.

В определенный момент возникает внутрикорпоративная конкуренция за ресурсы – как материальные, так и информационные. Защита и безопасность информации становятся важнейшей частью как существования и правильного функционирования системы в целом, так и индивидуального, заключающегося в оптимальном изменении и использовании ресурсов [2].

Время жизни и возникновение конкуренции

Образование, развитие и функционирование любой системы аналогично биологическим процессам в природе. Поэтому сравнение возможно, и далее будут использоваться термины «популяция», т.е. совокупность субъектов, и «индивидуум», собственно сам субъект. Каждый индивидуум имеет время жизни, в течение которого создает, перераспределяет и приобретает информацию, участвует в создании новых индивидуумов и защищенных образований. Время жизни корпорации больше, чем у индивида. Время жизни корпорации тратится на приобретение ресурсов и увеличение количества индивидов. Для лучшей конкуренции необходимо большее количество субъектов, соответственно корпорация растет. Как только численность индивидов в корпорации стабилизируется, конкуренция приобретает вид перераспределения и обработки информации.

Рассмотрим популяцию в ее наипростейшем виде, которая изображена на рис. 1. Имеется набор индивидуумов, каждый из которых контролирует какую-то часть ресурсов системы. На данном этапе все индивидуумы контролируют одинаковую часть системных ресурсов.

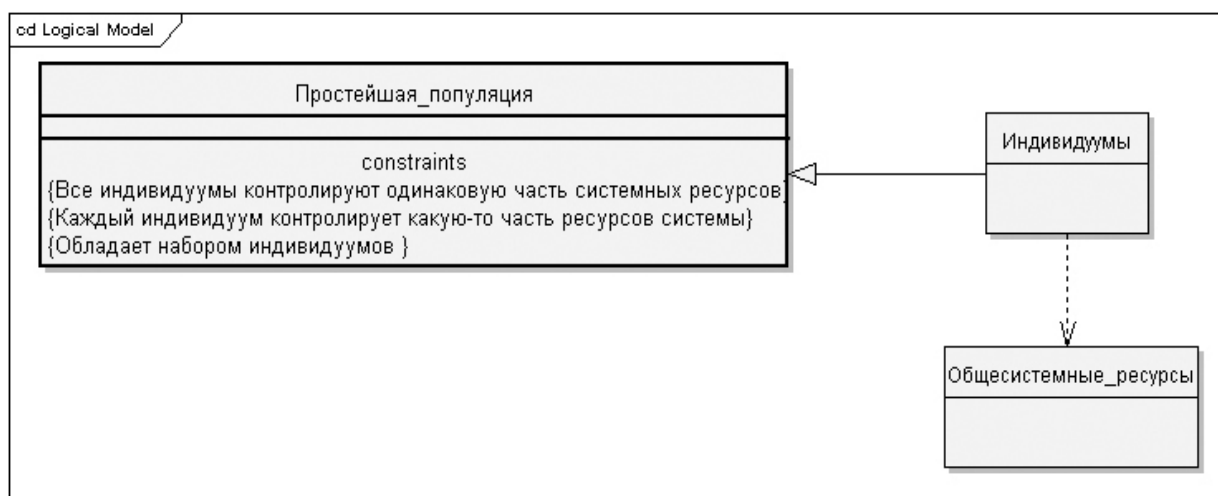


Рис. 1. Простейшая популяция

На данном этапе нет борьбы за распределение системных ресурсов между индивидуумами. При росте системы индивидуумы начинают пытаться овладеть одинаковой

частью системных ресурсов, т.е. появляется взаимодействие. Для взаимодействия между индивидуумами необходим инструмент, этим инструментом являются информация и язык. Все индивидуумы примерно равны, соответственно возникает необходимость в объединениях, которые будут обладать большими системными ресурсами, чем одиночные индивидуумы.

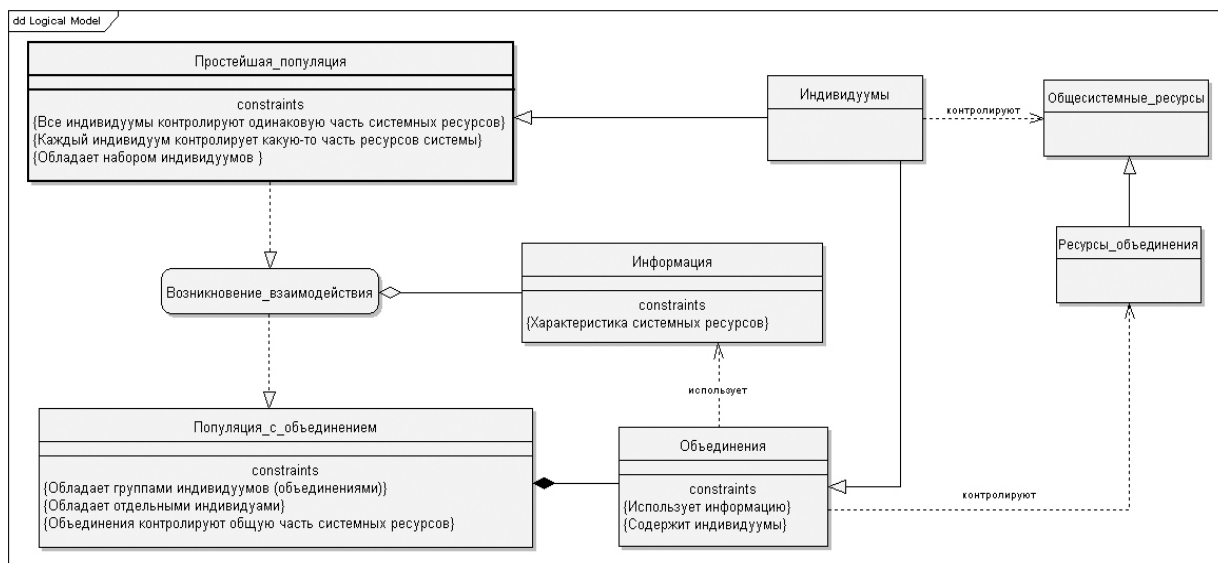


Рис. 2. Популяция с объединением

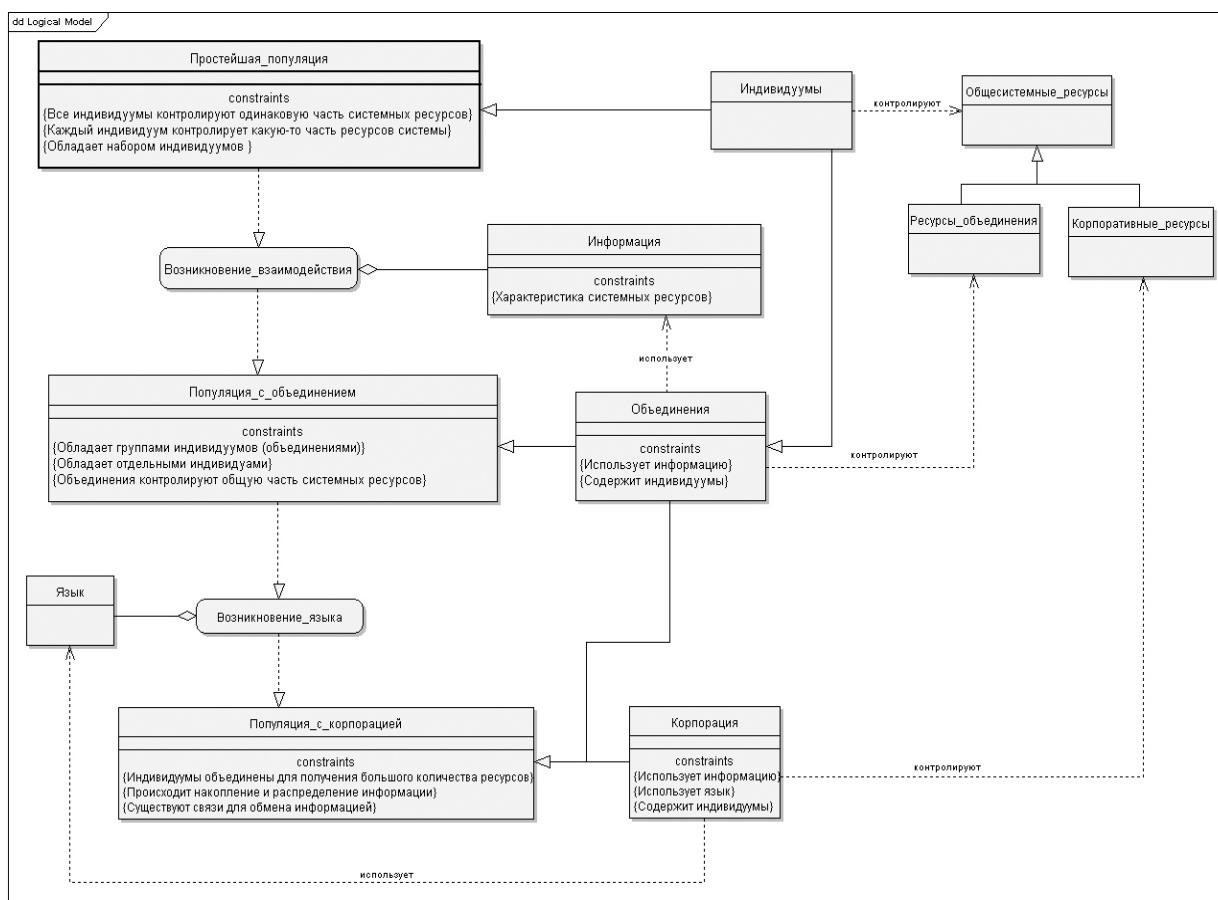


Рис. 3. Популяция с корпорацией

При наличии информации как инструмента борьбы за системные ресурсы возникают объединения, т.е. группы индивидуумов, контролирующих общую часть системных ресурсов. Здесь информация выступает в роли характеристики системных ресурсов, а, значит, помогает противостоять остальным индивидуумам популяции. При помощи информации индивидуум в объединении контролирует часть системных ресурсов объединения, которая заведомо больше, чем часть ресурсов, которые контролирует отдельный индивидуум (рис. 2).

После того, как образовались несколько объединений, необходим новый элемент для борьбы за ресурсы, которые или еще не распределены по объединениям, или за ресурсы самих объединений, причем между индивидуумами этих объединений. В этот момент и появляется язык, который способствует распределению информации между индивидуумами. Появление языка влечет за собой появление корпораций, в которых индивидуумы объединены для получения большого количества ресурсов и существуют связи для обмена информацией, т.е. происходит как само накопление информации, так и ее распределение. Популяция с корпорацией приведена на рис. 3.

В итоговой модели отражен процесс развития популяции, ее составляющие и связи между ними.

В корпорации каждый индивидуум при помощи языка устанавливает различные отношения с другими индивидуумами. В результате таких отношений возникает распределение информации внутри корпорации. Такая деятельность корпорации направлена на накопление корпоративных ресурсов и их сохранение. Так как информация – это инструмент для накопления ресурсов, то возникает необходимость в ее защите. Потеря информации, которой располагают индивидуумы, не может привести к большим потерям корпоративных ресурсов.

Языковые связи

Рассмотрим простейший пример взаимодействия внутри корпорации из трех индивидуумов, отображенной на рис. 4.

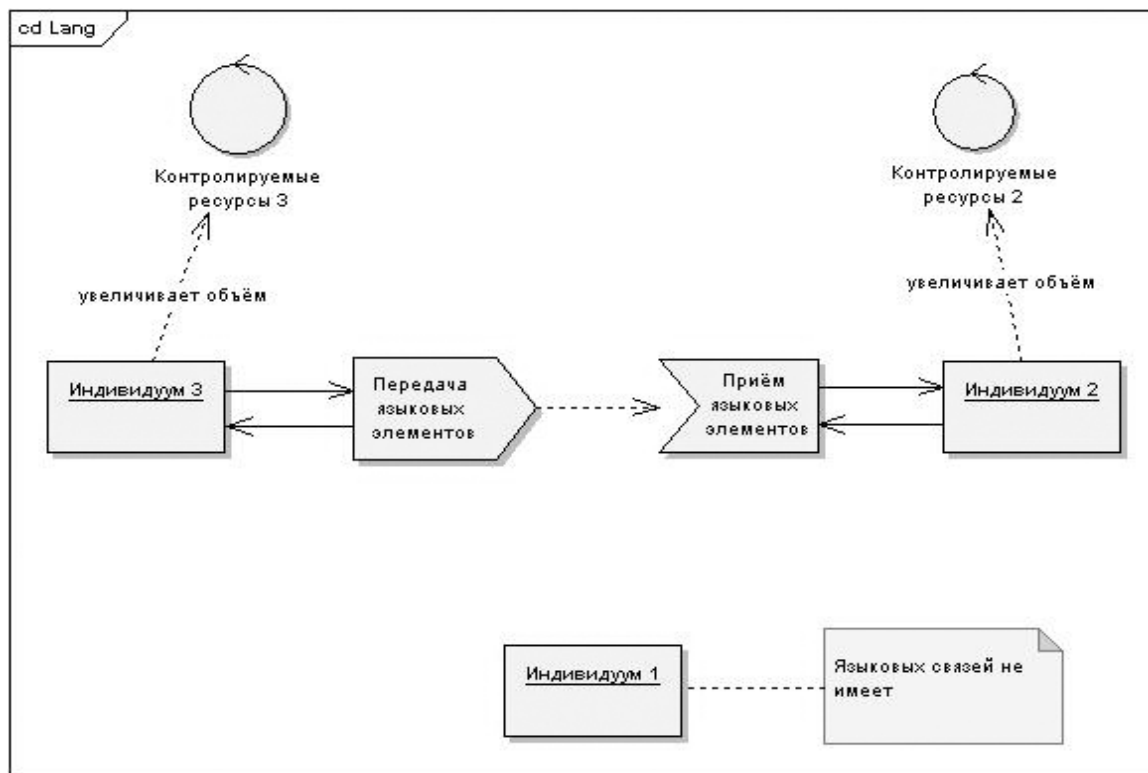


Рис. 4. Языковая связь в корпорации

Третий индивидуум устанавливает языковую связь со вторым и обменивается языковыми элементами. Этот обмен информацией направлен на увеличение объема контролируемых ресурсов, что происходит из-за получения новой информации. Здесь появляется понятие ценности информации – это величина корпоративной информации, которую теоретически можно использовать, относительно всей информации, располагаемой корпорацией. В данной ситуации ценность информации первого индивидуума равна нулю, поскольку он не участвует в языковых связях, а второго и третьего – $1/2$, так как они образуют корпорацию, состоящую из двух индивидуумов, внутри объединения.

Индивидуумы внутри корпорации разрабатывают набор языков. Эти языки используются очень долгое время по сравнению со временами жизни корпораций и индивидуумов. Языки развиваются вместе с корпорациями и сохраняются после исчезновения отдельных индивидуумов.

В разное или одно и то же время индивидуумы входят в состав разных корпораций. Поэтому одни и те же языки могут использоваться в разных связях и корпорациях.

Заключение

В статье проанализированы понятия корпорации, системы и ресурсов. Исследовался новый объект между субъектами корпоративной системы – многосубъектовое взаимодействие с использованием элементов тактической и стратегической конкуренции.

Корпоративную систему можно рассматривать как систему различных субъектов, между которыми происходят постоянные взаимодействия по обмену формализованными данными.

Различные по сложности и функциональным особенностям субъекты системы используют различные ресурсы, но некоторые ресурсы могут одновременно использоваться несколькими различными субъектами, в результате происходит борьба за ресурсы. Это фактор наличия конкурирующего взаимодействия между компонентами системы за контроль отдельных компонент над однородными общесистемными ресурсами. В процессе развития информационных технологий это привело к обострению вопросов защиты и безопасности информации.

Предложены структурные модели конкурентной борьбы субъектов корпоративной системы за ресурсы, рассмотрены принципы возникновения конкуренции и языковые связи между субъектами.

Литература

1. Осовецкий Л.Г., Немолочнов О.Ф., Твёрдый Л.В., Беляков Д.А. Основы корпоративной теории информации. – СПб: СПбГУИТМО, 2004. – С.54–69.
2. Берзин Е.А. Оптимальное распределение ресурсов и теория игр. – М.: Радио и связь, 1983. – 267 с.
3. Буч Г., Рамбо Д., Джекобсон А. Язык UML. Руководство пользователя: Пер. с англ. – М., ДМК, 2000. – 407с.
4. Шеннон К. Математическая теория связи. – М.: ИИЛ, 1963. – 193 с.

МЕТОДЫ ТЕСТИРОВАНИЯ ВЫЧИСЛИТЕЛЬНЫХ ПРОЦЕССОВ

О.Ф. Немолочнов, А.Г. Зыков, Л.Г. Осовецкий, В.И. Поляков

В работе предлагаются и рассматриваются примитивы вырожденных покрытий вершин графо-аналитических моделей программ, графо-аналитические модели двух ациклических вычислительных процессов, решающих одну и ту же задачу, построение их комплексных покрытий и два способа верификации: по покрытиям и перекрестным тестированием.

Введение

Постоянно повышающиеся требования к качеству программных продуктов заставляют разработчиков уделять все большее внимание технологическому процессу разработки, а также вопросам тестирования и верификации программ.

Разрабатываемая на факультете компьютерных технологий и управления кафедрами информатики и прикладной математики, вычислительной техники и безопасных информационных технологий автоматизированная учебно-исследовательская система верификации и тестирования (АУИСВ и Т) вычислительных процессов, в частном случае порождаемых программами при интерпретации их команд процессорами, базируется на графо-аналитических моделях (ГАМ) и комплексных кубических покрытиях булевых функций, обеспечивающих вычисления искомым переменных по тем или иным аналитическим выражениям произвольного вида. Тестирование вычислительных процессов является одним из методов их верификации [1].

Графо-аналитические модели

ГАМ задается в виде множества вершин: *условных*, описывающих ветвления по условиям предикатам, *линейных*, описывающих вычисление переменных по аналитическим формулам – выражениям, и *дуг* связи между ними. Сходящиеся дуги описываются с применением виртуальных вершин *объединения дуг*.

Комплексное кубическое покрытие состоит из логической части в виде покрытий и алгебраических выражений переменных. Оно строится по ГАМ на основе алгебро-топологических методов [2].

Примитивы покрытий вершин

Примитивные вырожденные покрытия вершин ГАМ показаны на рис.1.

Покрытия объединяют в себе булевы переменные $B=\{a, b, c, \dots\}$, вырабатываемые по условиям предикатам АЛУ процессора в виде специальных признаков – флагов, и реализуются командами условной передачи управления (*if Jump*) и сигналами управления $z \in Z$, вырабатываемыми ЦУУ и осуществляющими передачу управления от команды к команде.

Так как вершины ГАМ являются по определению замкнутыми множествами [3], то множества Z можно ограничить только переменными Z , осуществляющими передачу управления от вершине к вершине, т.е. убрать из рассмотрения внутренние передачи управления от команды к команде, чтобы не затенять и не усложнять излишними подробностями кубические покрытия. Заметим, что кубические покрытия являются строгим математическим описанием ГАМ любого вычислительного процесса, построенного из указанных выше трех типов вершин.

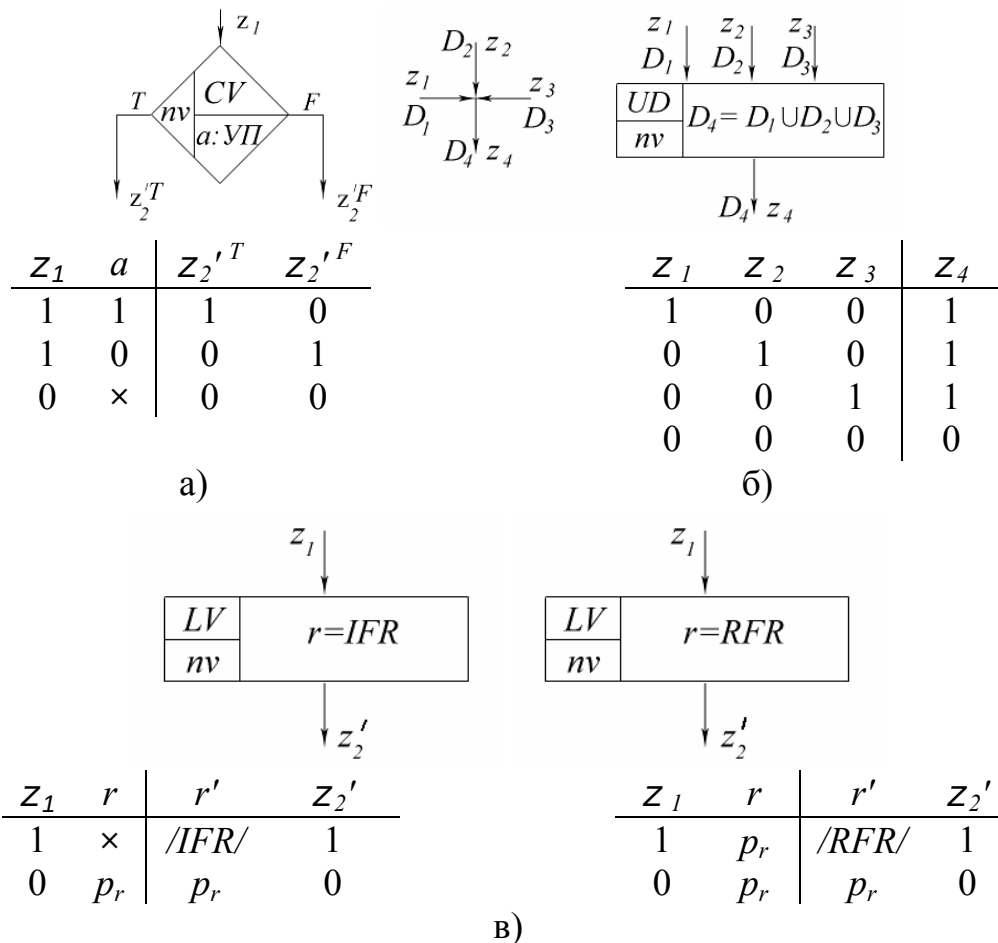


Рис.1. Прimitives и покрытия типовых вершин ГАМ: а) условной вершины (CV); б) объединения дуг (UD); в) линейной вершины (LV); УП – условие-предикат; nv – номер вершины; p_r – предыдущее значение переменной r ; $\times$ – безразличное значение переменной.

Комплексные покрытия переменных

Покрытия переменных $r \in R$, задаваемых согласно концептуальной модели вычислительного процесса [4] в виде итеративных и рекуррентных формул и выражений, строятся путем перебора путей по ГАМ от точки Tin до точки $Tout$ или наоборот и пересечения конкретных примитивных покрытий вершин, входящих в рассматриваемый путь. Если в процессе перебора путей встречаются противоречия, т.е. кубы из примитивов дают пустое пересечение, то данный куб просто удаляется из покрытия.

Методы верификации

Общая схема верификации показана на рис.2.

Здесь рассмотрен случай, когда один и тот же вычислительный процесс реализован двумя различными способами. На схеме показано, что на основании технических заданий (ТЗ) или разработанных программ строятся ГАМ и комплексные покрытия двух рассматриваемых вычислительных процессов. Из них исключаются значения, на которых функция не определена (*don't care*), методом алгебро-топологического вычитания, и строятся контролирующие тесты. Предложены два способа верификации этих процессов:

- метод алгебро-топологического вычитания покрытий каждого из каждого. При условии пустого значения результата делается заключение о эквивалентности данных вычислительных процессов;
- метод построения тестовых наборов по комплексным покрытиям путем пересечения кубов из интервальных частей покрытий и перекрестного тестирования, по результатам которого делается заключение о результатах верификации.

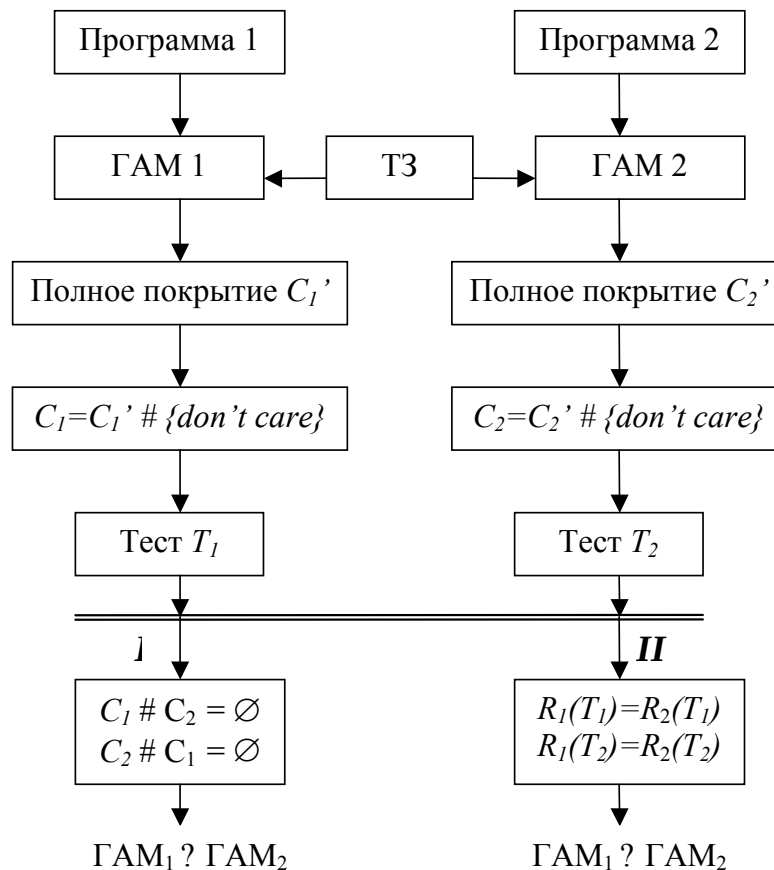


Рис. 2. Общая схема процесса верификации вычислительных процессов

Различия вычислительных процессов могут порождаться:

- различными покрытиями одних и тех же множеств предикатов;
- различием их функциональной декомпозиции, что может быть обусловлено выбором начального условия;
- проектированием программ на различных алгоритмических языках и различными трансляторами;
- различием аналитических выражений для вычисляемых переменных, формулы для которых подвергнуты преобразованиям с целью их упрощения или, например, с целью минимизации погрешности вычислений.

При проведении верификации по покрытиям следует учитывать следующие факторы:

- компенсацию условий предикатов;
- эквивалентность, т.е. размножение некоторых условий;
- несуществующие значения (*don't care*), возникающие в ходе функциональной декомпозиции неполностью определенных функций.

Указанные факторы вызывают основные затруднения в алгоритмизации и построении САПР верификации.

Пример верификации ациклического процесса

Пусть задана некоторая интервальная формула:

$$r = \begin{cases} FR1, & \text{при } x \leq k_1; \\ FR2, & \text{при } k_1 < x < k_2; \\ FR3, & \text{при } x \geq k_2, \end{cases}$$

реализующая вычисления некоторой переменной r по различным формулам: $FR1$, $FR2$ и $FR3$ произвольного вида в зависимости от двух булевых переменных, задающих некоторые условия предикаты в виде неравенств: $a: x \leq k_1$; $b: x \geq k_2$.

Переход от неравенств к булевым переменным при проектировании вычислительного процесса позволяет абстрагироваться от конкретного смысла неравенств и соответствующих им условий предикатов и рассмотреть решение задачи верификации в общем виде. Графо-аналитические модели вычисления переменной r приведены на рис. 3.

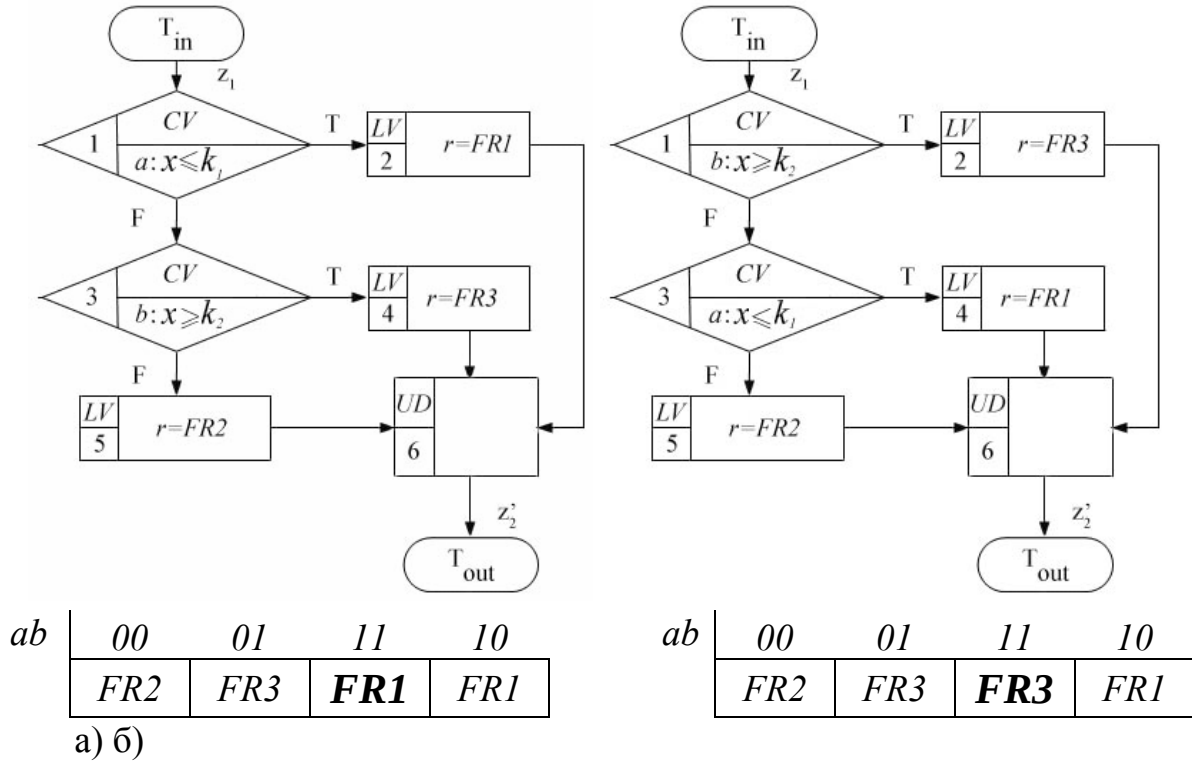


Рис. 3. ГАМ вычисления интервальной формулы

На рис. 3а показана функциональная декомпозиция булевой функции $f=f(a,b)$ с начальной вершиной условия предиката a (ГАМ 1), а на рис. 3б – с начальной вершиной b (ГАМ 2).

Построим комплексные кубические покрытия $C_1(r)$ и $C_2(r)$ для вычисляемой переменной r по ГАМ 1 и ГАМ 2:

$$C_1(r) = \left\{ \begin{array}{c|c|c|c|c|c|c} z_1 & a & b & r & r' & z_2' & \{c\} \\ \hline 1 & 1 & \times & \times & /FR1/ & 1 & c_1 \\ 1 & 0 & 1 & \times & /FR3/ & 1 & c_2 \\ 1 & 0 & 0 & \times & /FR2/ & 1 & c_3 \\ 0 & \times & \times & p & p & 0 & c_4 \end{array} \right.$$

$$C_2(r) = \left\{ \begin{array}{c|c|c|c|c|c|c} z_1 & a & b & r & r' & z_2' & \{c\} \\ \hline 1 & 1 & \times & \times & /FR1/ & 1 & c_1 \\ 1 & 0 & 1 & \times & /FR3/ & 1 & c_2 \\ 1 & 0 & 0 & \times & /FR2/ & 1 & c_3 \\ 0 & \times & \times & p & p & 0 & c_4 \end{array} \right.$$

По покрытиям $C_1(r)$ и $C_2(r)$ построим тесты $T_1(r)$ и $T_2(r)$ путем пересечения кубов из интервальных частей покрытий $C_1(r)$ и $C_2(r)$, соответственно. Получим тесты:

$$T_1(r) = \left\{ \begin{array}{c|c|c} \begin{array}{cccc} z_1 & a & b & r \end{array} & \begin{array}{cc} r' & z_2' \end{array} & \{t\} \\ \hline \begin{array}{cccc} 1 & 1' & 0 & \times \\ 1 & 0' & 0 & \times \\ 1 & 0 & 1' & \times \\ 1 & 0 & 0' & \times \\ 1' & 0 & 1 & p \\ 0' & 0 & 1 & p \end{array} & \begin{array}{cc} /FR1/ & 1 \\ /FR2/ & 1 \\ /FR3/ & 1 \\ /FR2/ & 1 \\ /FR3/ & 1 \\ p & 0 \end{array} & \begin{array}{c} t_1 \\ \tilde{t}_1 \\ t_2 \\ \tilde{t}_2 \\ t_3 \\ \tilde{t}_3 \end{array} \end{array} \right. \quad T_2(r) = \left\{ \begin{array}{c|c|c} \begin{array}{cccc} z_1 & a & b & r \end{array} & \begin{array}{cc} r' & z_2' \end{array} & \{t\} \\ \hline \begin{array}{cccc} 1 & 0 & 1' & \times \\ 1 & 0 & 0' & \times \\ 1 & 1' & 0 & \times \\ 1 & 0' & 0 & \times \\ 1' & 1 & 0 & p \\ 0' & 0 & 1 & p \end{array} & \begin{array}{cc} /FR3/ & 1 \\ /FR2/ & 1 \\ /FR1/ & 1 \\ /FR2/ & 1 \\ /FR1/ & 1 \\ p & 0 \end{array} & \begin{array}{c} t_1 \\ \tilde{t}_1 \\ t_2 \\ \tilde{t}_2 \\ t_3 \\ \tilde{t}_3 \end{array} \end{array} \right.$$

Примеч. $t_1/\tilde{t}_1 \in c_1 \cap c_3$ из $C_1(r)$

Примеч. $t_1/\tilde{t}_1 \in c_1 \cap c_3$ из $C_2(r)$

Заметим, что для формул должны выполняться условия: $/FR1/ \neq /FR2/ \neq /FR3/ \neq p$, т.е. вычисляемые и хранимые значения должны различаться на разных наборах теста. Штрихами в тестах отмечены значения активно изменяемых условий-предикатов.

С учетом удаления $ab=11$ верификация по покрытиям дает $C_1 \# C_2 = \emptyset$ и $C_2 \# C_1 = \emptyset$, что и свидетельствует об эквивалентности вычислительных процессов. Это наглядно можно наблюдать на картах Карно, приведенных на рис.3.

Перекрестное тестирование дает следующий результат: $R_1(T_1)=R_2(T_1)$ и $R_1(T_2)=R_2(T_2)$, что также подтверждает эквивалентность вычислительных процессов. Заметим, что если переменную r вычислять по разным упрощенным формулам $FR1$, $FR2$ и $FR3$, то метод перекрестного тестирования является предпочтительным, так как не требует приведения выражений формул к каноническому виду.

Заключение

Рассмотренные методы верификации вычислительных процессов с использованием покрытий и перекрестного тестирования являются алгоритмической основой построения САПР логических схем и программного продукта, в том числе и АУИСВ и Т.

Литература

1. Липаев В.В. Сопровождение и управление сложных программных средств. – М.: СИНТЕГ, 2006. – 372 с.
2. Проектирование цифровых вычислительных машин / С.А. Майоров, Г.И. Новиков, О.Ф. Немолочнов и др. Под ред. С.А. Майорова. – М.: Высш. шк., 1972. – 344 с.
3. Немолочнов О.Ф., Зыков А.Г., Поляков В.И. Комплексные кубические покрытия и графо-аналитические модели как средство описания вычислительных процессов программ /Труды Международных научно-технических конференций «Интеллектуальные системы» (AIS'06) и «Интеллектуальные САПР» (CAD-2006), Научное издание в 3-х т. Т. 2. – М.: Физматлит, 2006. – С. 3–7.
4. Итерационно-рекурсивная модель вычислительных процессов программ / Немолочнов О.Ф., Зыков А.Г., Поляков В.И., Осовецкий Л.Г., Сидоров А.В. Кулагин В.С. // Известия вузов. Приборостроение. 2005. Том 48. № 12. С.14–20.

**ТЕХНОЛОГИЯ ПРОВЕДЕНИЯ ЕГЭ ПО ИНФОРМАТИКЕ
В КОМПЬЮТЕРНОМ ВИДЕ****В.Н. Васильев, Л.С. Лисицына, А.В. Лямин**

В статье изложена концепция разработки сетевой информационной системы (ИС) для проведения экзаменов по информатике у школьников выпускных классов в формате единого государственного экзамена (ЕГЭ) в компьютерном виде. Представлены результаты разработки и апробации содержания экзаменационной работы, технологий тестирования и шкалирования результатов испытаний в ИС, программно-аппаратного комплекса системы на примере проведения в феврале 2007 г. олимпиады по информатике у школьников выпускных классов г. Санкт-Петербурга.

Введение

Единый государственный экзамен (ЕГЭ) призван оценить подготовку выпускников общеобразовательных школ по основным разделам школьных предметов с целью зачисления в образовательные учреждения среднего и высшего профессионального образования. Опыт проведения ЕГЭ по информатике в 2004–2006 гг. в различных регионах РФ показал, что существенным недостатком этого экзамена является «бланковая технология» его проведения [1]. Следует заметить, что использование компьютеров при аттестации школьников в рамках проведения ЕГЭ по информатике позволит оценить навыки работы школьников на персональных компьютерах, а использование сетевых технологий для его проведения – навыки использования школьниками интернет-технологий, что предусматривается при изучении этого школьного предмета.

Кроме того, в рамках предлагаемой концепции предусматривается разработка виртуальной лаборатории, которая позволит, с одной стороны, продемонстрировать школьнику знания и умения написания и отладки программ, а, с другой стороны, проверить эксперту ЕГЭ по информатике работоспособность этой программы. Разработка технологий проведения экспертизы в среде виртуальной лаборатории позволит повысить объективность оценивания и снизить трудоемкость проверки экзаменационных работ.

В 2007 г. о своем участии в ЕГЭ по информатике заявили 13 регионов РФ. В дальнейшем с учетом возрастающего значения информационно-коммуникационных технологий (ИКТ) роль информатики будет возрастать, увеличится и количество регионов РФ, которые будут проводить ЕГЭ по информатике. Поэтому переход к ЕГЭ по информатике в компьютерном виде является весьма актуальной и своевременной проблемой.

Кроме того, проводимые в СПбГУ ИТМО университетские и региональные олимпиады по информатике, направленные на поиск талантливой, наиболее подготовленной молодежи к обучению в вузах Санкт-Петербурга, должны, в том числе, служить подготовке школьников к ЕГЭ по информатике, поэтому должны проводиться в формате ЕГЭ.

**Структура и содержание экзаменационной работы по информатике
в компьютерном виде**

За основу разработки были взяты содержание и демо-версия экзаменационной работы Федеральной предметной комиссии ЕГЭ по информатике в 2007 г., опубликованные на сайте Федерального института педагогических измерений (ФИПИ)

<http://fipi.ru/view/sections/157/docs/>, с сохранением структуры и основных характеристик работы, но с сокращением времени тестирования с 240 мин. до 130 мин. за счет сокращения количества заданий в работе на 50%. При разработке демо-версии ЕГЭ по информатике в компьютерном виде учитывалось не только сокращение заданий в экзаменационной работе, но и возможности технологий компьютерного тестирования сетевой ИС.

Структура экзаменационной работы по информатике в компьютерном виде полностью повторяет распределение заданий по частям работы (табл. 1), по уровням сложности (табл. 2), по видам проверяемой деятельности (табл. 3) в ЕГЭ по информатике 2007 г. Перераспределение заданий по разделам курса информатики (табл. 4) выполнено таким образом, чтобы в первую очередь проверялись навыки алгоритмизации и технологий программирования, что отвечает требованиям вузов к поступающим на профильные специальности. В табл. 5 приведен план экзаменационной работы.

Экзаменационная работа в компьютерном виде в формате ЕГЭ по информатике состоит из трех частей. Часть 1 (А) содержит 10 заданий базового и повышенного уровня сложности. В этой части собраны задания с выбором ответа, подразумевающие выбор одного правильного ответа из четырех предложенных. Задания выполняются и проверяются в системе тестирования. Часть 2 (В) содержит 4 задания базового и повышенного уровней сложности. В этой части собраны задания с краткой формой ответа, подразумевающие самостоятельное формулирование и ввод ответа. Задания выполняются и проверяются в системе тестирования. Часть 3 (С) содержит 2 задания высокого уровня сложности. Задания этой части выполняются в среде виртуальной лаборатории системы тестирования.

Содержание заданий разработано по основным темам курса информатики и информационных технологий, объединенных в следующие тематические блоки: «Информация и ее кодирование», «Алгоритмизация и программирование», «Основы логики», «Моделирование и компьютерный эксперимент», «Программные средства и технологии ИКТ», «Технологии программирования».

Содержанием экзаменационной работы охватывается основное содержание курса информатики, важнейшие его темы, наиболее значимый в них материал, однозначно трактуемый в большинстве преподаваемых в школе вариантов курса информатики. Часть 1 содержит задания из всех тематических блоков, кроме заданий тем «Программные средства и технологии ИКТ» и «Технологии программирования». Часть 2 включает задания по темам «Основы логики», «Алгоритмизация и программирование», «Программные средства и технологии ИКТ», «Технологии программирования». Задания части 3 направлены на проверку сформированности важнейших умений разработки алгоритмов, написания и отладки программ, предусмотренных требованиями к обязательному уровню подготовки по информатике учащихся средних общеобразовательных учреждений. Эти умения проверяются на высоком уровне сложности по следующим темам информатики: «Алгоритмизация и программирование» и «Технологии программирования».

На уровне *представлений* проверяется материал о программных средствах и технологиях ИКТ, поскольку изучаемые технологии разнообразны и используют различные версии и платформы.

На уровне *воспроизведения знаний* проверяется такой теоретический материал, как:

- единицы измерения информации;
- принципы кодирования;
- моделирование как метод научного познания.

Материал на проверку сформированности *умений применять свои знания в стандартной ситуации* входит во все три части экзаменационной работы. Здесь проверяются следующие умения:

- осуществлять перевод из одной системы счисления в другую;
- осуществлять арифметические действия в двоичной, восьмеричной и шестнадцатеричной системах счисления;
- использовать стандартные алгоритмические конструкции при программировании;
- формально исполнять алгоритмы, записанные на естественных и алгоритмических языках, в том числе на языках программирования;
- создавать и преобразовывать логические выражения;
- формировать для логической функции таблицу истинности.

Материал на проверку сформированности *умений применять свои знания в новой ситуации* входит во все три части экзаменационной работы. Здесь проверяются следующие умения:

- подсчитывать информационный объем сообщения;
- решать логические задачи;
- анализировать текст программы с точки зрения соответствия записанного алгоритма поставленной задаче и находить в ней ошибки;
- реализовывать сложный алгоритм с использованием приемов написания и отладки программ, записанных в стиле наиболее распространенных языков программирования.

Часть 1 (А) содержит задания, которые относятся к базовому и повышенному уровню сложности. Часть 2 (В) содержит по два задания базового и повышенного уровней сложности. Задания части 3 (С) относятся к высокому уровню. Предполагаемый процент выполнения заданий базового уровня 60–90%. Предполагаемый процент выполнения заданий повышенного уровня 40–60%. Предполагаемый процент выполнения заданий части С – менее 40%. Для оценки достижения базового уровня используются задания с выбором ответа и кратким ответом. Достижение уровня повышенной подготовки проверяется с помощью заданий с выбором ответа, кратким и развернутым ответом. Для проверки достижения высокого уровня подготовки в экзаменационной работе используются задания с кратким и развернутым ответом.

Части работы	Число заданий	Максимальный первичный балл	Процент максимального первичного балла	Тип заданий	Время выполнения
Часть 1	10	10	50	ВО	25
Часть 2	4	4	20	КО	15
Часть 3	2	6	30	РО	90
Итого:	16	20	100%		130

Табл. 1. Распределение заданий по частям экзаменационной работы

Уровень сложности заданий	Число заданий	Максимальный первичный балл	Процент максимального первичного балла	Перечень заданий
Базовый	8	8	40	A1.A3.A4.A5.A7.A9.B2.B3
Повышенный	6	7	35	A2.A6.A8.A10.B1.B4
Высокий	2	5	25	C1.C2
Итого:	16	20	100	

Табл. 2. Распределение заданий по уровням сложности

Код	Виды деятельности	Число заданий	Максимальный первичный балл	Процент максимального первичного балла	Перечень заданий
1	Воспроизведение представлений или знаний	3	3	15	A1.A7.B3
2	Применение знаний и умений в стандартной ситуации	8	9	45	A3.A4.A5.A8.A9.B2.B4.C1
3	Применение знаний и умений в новой ситуации	5	8	40	A2.A6.A10.B1.C2
	Итого:	16	20	100	

Табл. 3. Распределение заданий по видам проверяемой деятельности

№ п/п	Название раздела	Число заданий	Максимальный первичный балл	Процент максимального первичного балла	Перечень заданий
1	Информация и ее кодирование	4	4	20	A1.A2.A3.A10
2	Алгоритмизация и программирование	5	6	30	A4.A5.A9.B2.C1
3	Основы логики	3	3	15	A6.A8.B4
4	Моделирование и компьютерный эксперимент	1	1	5	A7
5	Программные средства и технологии ИКТ	1	1	5	B3
6	Технология программирования	2	5	25	B1.C2
	Итого:	16		100	

Табл. 4. Распределение заданий по разделам курса информатики

№	Обозначение задания в работе	Проверяемые элементы содержания	Коды проверяемых элементов содержания по кодификатору*	Коды видов деятельности	Уровень сложности задания	Макс. балл за выполнение задания	Примерное время выполнения задания (мин.)
		Часть 1					
1	A1	Кодирование текстовой информации. Кодировка ASCII. Основные используемые кодировки кириллицы.	1.1.8.	1	Б	1	1

2	A2	Умение подсчитывать информационный объем сообщения	1.1.4.	3	П	1	2
3	A3	Умения выполнять арифметические операции в двоичной, восьмеричной и шестнадцатеричной системах счисления	1.1.7.	2	Б	1	2
4	A4	Знание и умение использовать основные алгоритмические конструкции: следование, ветвление, цикл	1.2.2.	2	Б	1	1
5	A5	Использование переменных. Объявление переменной (тип, имя, значение). Локальные и глобальные переменные.	1.2.3.	2	Б	1	2
6	A6	Знание основных понятий и законов математической логики	1.3.1.	3	П	1	2
7	A7	Умение представлять и считывать данные в разных типах информационных моделей (схемы, карты, таблицы, графики и формулы)	1.4.2.	1	Б	1	2
8	A8	Умение строить и преобразовывать логические выражения	1.3.2.	2	П	1	8
9	A9	Формальное исполнение алгоритма, записанного на естественном языке	1.2.1.	2	Б	1	2
10	A10	Умение определять скорость передачи информации при заданной пропускной способности канала	1.1.6./ 2.7.1	3	П	1	3
		Часть 2					
11	B1	Умение прочесть фрагменты программы на языке программирования и найти допущенные ошибки	2.8.1./ 2.8.2.	3	П	1	5
12	B2	Умение исполнять алгоритм в среде формального исполнителя	1.2.2.	2	Б	1	6
13	B3	Знания о файловой системе организации данных, о технологиях создания и обработки информации в электронных таблицах, базах данных и графических редакторах	2.2.3./ 2.4.1./ 2.5.2./ 2.6.1.	1	Б	1	1
14	B4	Умения строить таблицы истинности и логические схемы	1.3.3.	2	П	1	3
		Часть 3					
15	C1	Умения написать короткую (10-15 строк) простую программу обработки массива	1.2.4/ 2.8.3	2	В	2	30
16	C2	Умения создавать собственные программы (30-50 строк) для решения задач средней сложности	2.8.3.	3	В	4	60
Всего заданий – 16, из них по типу заданий: А – 10, В – 4, С – 2; по уровню сложности: Б – 8, П – 6, В – 2. Максимальный первичный балл за работу – 20. Общее время выполнения работы – 130 мин.							

* - кодификатор содержания, см. <<http://fipi.ru/view/sections/157/docs/>>.

Табл. 5. План экзаменационной работы

Технологии компьютерного тестирования для заданий части 1 (А)

Для компьютерного тестирования по заданиям части 1 в ИС используется технология заданий закрытой формы. На рис. 1 представлен пример тестирования в ИС по одному из заданий этой части.

The screenshot shows the ANTP testing system interface. At the top, it displays 'Количество кадров: 14' and 'Количество заданий: 14'. Below this, a question is presented: 'Значение выражения $10_{16} + 10_8 * 10_2$ в двоичной системе счисления равно:'. There are four radio button options: 'a 1010', 'b 11010', 'c 110000', and 'd 100000'. At the bottom, there is a progress bar with 14 frames, a timer showing '02:03:25', and buttons for 'Предыдущий', 'Следующий', and 'Завершить'.

Рис. 1. Технология тестирования для заданий части 1 (А)

Технологии компьютерного тестирования для заданий части 2 (В)

Для компьютерного тестирования по заданиям части 2 в ИС применяются различные технологии заданий открытой формы: на рис. 2 приведен пример тестового задания В1, на рис. 3 – задания В2, на рис. 3 – задания В3, на рис. 4 – задания В4. Следует заметить, что применяемые здесь технологии для заданий В2 и В4 были специально разработаны для проведения экзамена и являются оригинальными.

The screenshot shows the ANTP testing system interface for a programming task. It displays 'Количество кадров: 14' and 'Количество заданий: 14'. The task description is: 'Требовалось написать программу, которая решает уравнение "ax+b=0" относительно x для любых чисел a и b, введенных с клавиатуры. Все числа считаются действительными. Программист торопился и написал программу неправильно. Исправьте допущенные ошибки. В поле ввода, расположенном ниже, укажите через пробел номера строк, в которых могут произойти ошибки во время их трансляции и выполнения:'. Below the text is a table with three columns: 'ПРОГРАММА НА ПАСКАЛЕ', 'ПРОГРАММА НА БЕЙСИКЕ', and 'ПРОГРАММА НА СИ'. Each column contains a snippet of code. At the bottom, there is a progress bar with 14 frames, a timer showing '02:00:31', and buttons for 'Предыдущий', 'Следующий', and 'Завершить'.

ПРОГРАММА НА ПАСКАЛЕ	ПРОГРАММА НА БЕЙСИКЕ	ПРОГРАММА НА СИ
<pre>1 var 2 a, b: Real; 3 begin 4 ReadLn(a,b); 5 if b = 0 else 6 Write('x = 0') 7 then 8 if a < 0 then 9 Write('нет решений') 10 else 11 Write('x =', -b/a); 12 end.</pre>	<pre>1 REM 2 REM 3 REM 4 INPUT a, b 5 IF b = 0 ELSE 6 PRINT "x = 0" 7 THEN 8 IF a < 0 THEN 9 PRINT "нет решений" 10 ELSE 11 PRINT "x =", -b/a 12 ENDIF 13 ENDIF 14 END</pre>	<pre>1 void main(void) { 2 float a,b; 3 4 scanf("%f%f", &a,&b); 5 if (b==0) else 6 printf("x=0"); 7 then 8 if (a<0) 9 printf("нет решений"); 10 else 11 printf("x=%f", -b/a); 12 }</pre>

Рис. 2. Технология тестирования для задания В1

Количество кадров: 14
Количество заданий: 14 Количество ответов: 2

Чертежный автомат предназначен для выполнения рисунков, содержащих горизонтальные и вертикальные линии, и может выполнять следующие четыре команды:

0 – поднять пишущий инструмент (ПИ) и установить в начало координат
1 – поднять ПИ
2 – опустить ПИ
(Δ x, Δ y) – сместить ПИ на Δ x–клеток по оси OX и Δ y–клеток по оси OY

Выполните следующую программу чертежного автомата: 0 2 (2,0) (0,3) (2,0) (0, -1) 0

Результаты выполнения программы укажите последовательными щелчками мыши в центрах раскрашенных автоматом клеток на поле.

8									
7									
6									
5									
4									
3			x	x	x				
2			x		x				
1			x						
0	x	x	x						
	0	1	2	3	4	5	6	7	8

Время на тест: 01:55:56
Номер кадра:

1 2 3 4 5 6 7 8 9 10 11 12 13 14

Предыдущий

Следующий

Завершить

Рис. 3. Технология тестирования для задания В2

Количество кадров: 14
Количество заданий: 14 Количество ответов: 3

Поставьте в соответствие понятия и их определения:

1. База данных	2. Поименованная область данных во внешней памяти компьютера.
2. Файл	3. Информация, задающая изображение по точкам.
3. Растровая графика	1. Совокупность определенным образом организованной информации, позволяющая упорядоченно хранить данные о группе объектов, обладающих одинаковыми свойствами.
4. Электронная таблица	4. Табличное представление исходных данных и результатов, а также связей между ними.

Время на тест: 01:52:38
Номер кадра:

1 2 3 4 5 6 7 8 9 10 11 12 13 14

Предыдущий

Следующий

Завершить

Рис. 4. Технология тестирования для задания В3



Количество кадров: 14
Количество заданий: 14 Количество ответов: 4

Заполните таблицу истинности выражения F от трех аргументов: X, Y, Z.

X	Y	Z	F = X ∧ Y ∧ ¬Z
0	1	0	0
1	1	0	1
1	0	1	

Время на тест: 01:49:55

Номер кадра:

1 2 3 4 5 6 7 8 9 10 11 12 13 14

Предыдущий

Следующий

Завершить

Рис. 5. Технология тестирования для задания В4

Виртуальная лаборатория для части С

Разработка виртуальной лаборатории ИС проводилась на основе реализации следующих *принципов*.

- Виртуальная лаборатория должна позволять школьнику продемонстрировать, а эксперту проверить навыки написания и отладки программ.
- Школьнику, чтобы пройти тестирование на умение применять технологии программирования, не потребуется что-то дополнительно изучать.
- Интерфейс лаборатории должен быть интуитивно понятен для школьника, имеющего опыт написания и отладки программ.
- Использовать основные стили программирования, изучаемые в школах на уроках информатики – Basic, Pascal и C (ввод, отладка и проверка работоспособности обеспечивается лабораторией), а также «свободный» стиль (для ввода текста программы, написанной на любом языке).

Аплет виртуальной лабораторной работы по информатике позволяет задать переменные, необходимые для работы программы, написать код программы, выбрать стиль кодирования и использовать готовые конструкции выбранного стиля. Корректно написанные программы можно отлаживать в пошаговом режиме или выполнять с возможностью остановки. Общий вид интерфейса виртуальной лаборатории представлен на рис. 6. Основное окно лаборатории предназначено для ввода исходного кода в выбранном стиле кодирования. Справа – окно для выбора операторов и операций выбранного стиля кодирования. Сверху – окна для представления входных, выходных и внутренних переменных программы, заполняемых в режиме редактирования (рис. 7). Окно «Сообщения» – для диагностических сообщений в режиме отладки и выполнения программы.

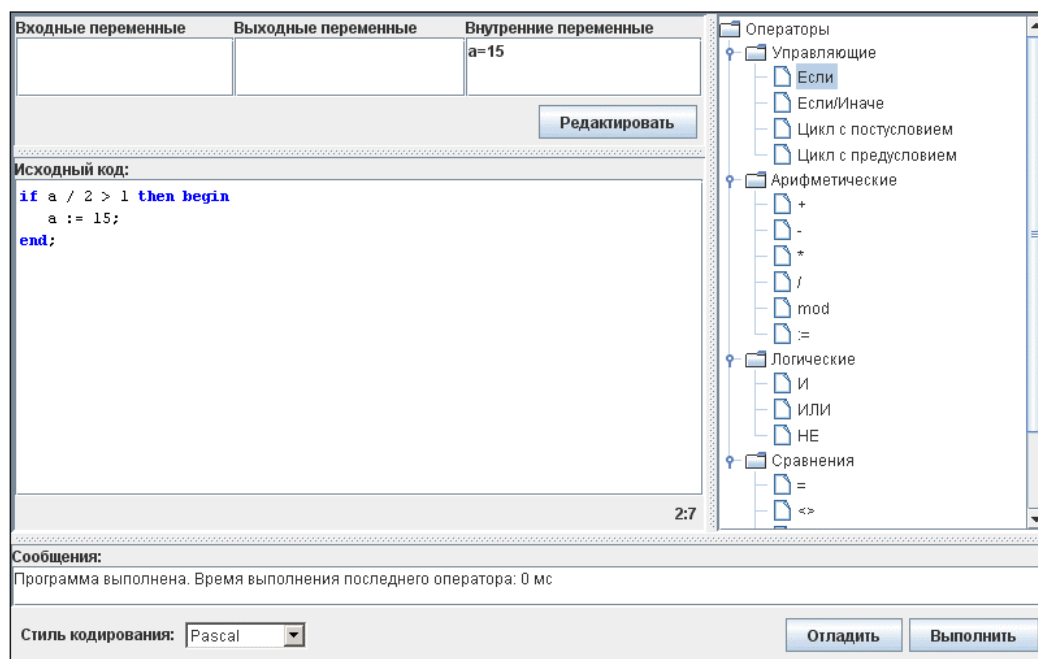


Рис. 6. Интерфейс виртуальной лаборатории ИС

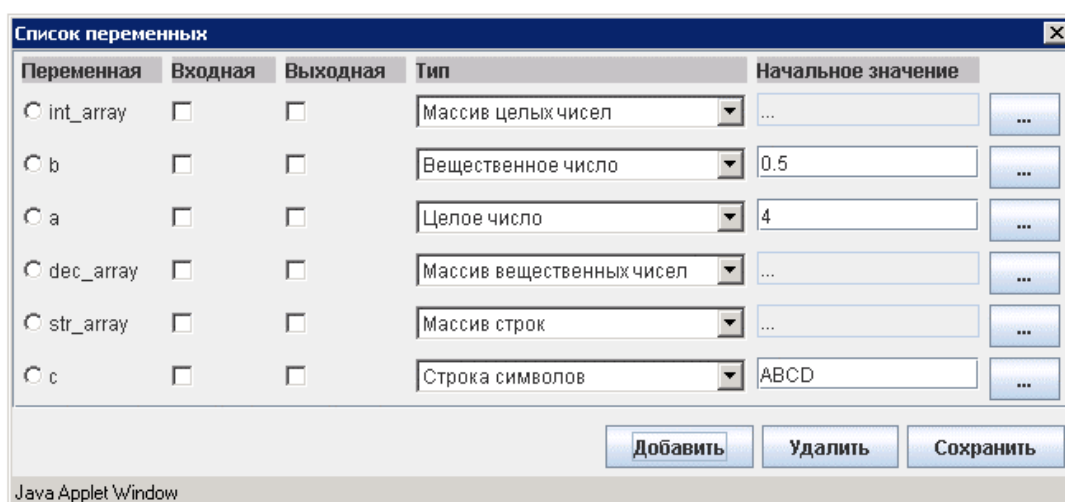


Рис. 7. Редактирование переменных в виртуальной лаборатории

Критерии оценивания заданий части С

В табл. 6 приведены критерии для оценивания экспертами задания С1, а в табл. 7 – задания С2.

Указания по оцениванию С1	Баллы
Предложен правильный алгоритм и написана работоспособная программа.	2
Предложен правильный алгоритм, но написана неработоспособная программа.	1
Предложен неправильный алгоритм.	0
Максимальный балл	2

Табл. 6. Критерии оценивания задания С1

Указания по оцениванию С2	Баллы
Создана работоспособная программа, реализующая правильный алгоритм решения задачи.	4
Создана работоспособная программа, но в алгоритме допущены ошибки, устраняемые за счет не более 2-ух исправлений.	3
Создана неработоспособная программа, реализующая правильный алгоритм решения задачи. Работоспособность программы установлена экспертом за счет не более 2-ух исправлений.	2
Создана неработоспособная программа с ошибками в алгоритме, но с демонстрацией навыков программирования.	1
Задание выполнено неверно	0
Максимальный балл	4

Табл. 7. Критерии оценивания задания С2

Сценарии тестирования в ИС

В основу разработки сценариев компьютерного тестирования заложены следующие принципы.

- Школьник получает экзаменационную работу и может выбирать для выполнения задания в произвольном порядке.
- Должна быть возможность исправления ответа до момента завершения тестирования.
- Время, использованное на тестирование, должно складываться из времени, затраченного на выполнение каждого задания (начало отсчета – с момента появления, а конец – в момент отсылки задания).

Сценарии проверки экзаменационных работ в ИС

В основу разработки сценариев проверки экзаменационных работ заложены следующие принципы.

- Автоматическая проверка частей А и В (без участия экспертов).
- Задания части С проверяются двумя независимыми экспертами (интернет-экзамен позволяет привлекать экспертов из различных регионов) и направляется старшему эксперту, если есть разногласие в оценках двух экспертов.
- Автоматизированная и формализованная проверка заданий части С экспертом в среде виртуальной лаборатории.

Методика шкалирования результатов экзамена

1. Для оценки каждого i -го задания – O_i вводятся следующие коэффициенты:
 W_i – вес задания, равный максимальному первичному баллу задания (по плану работы);
 C_i – степень выполнения задания:

- для части А – 0 (выполнено не верно) или 1 (выполнено верно);
- для части В – числовое значение из интервала от 0 до 1, устанавливаемое системой на основе анализа полноты ответа, например: $C_i = 0.5$ для ответа на задание В4 (заполнение таблицы истинности), содержащего 50% правильных значений таблицы;

- для части С – числовое значение из интервала от 0 до 1, устанавливаемое экспертом по критериям заданий C1 (0, 0.5, 1) и C2 (0, 0.25, 0.5, 0.75, 1).

S_i – вес уровня сложности задания:

- для заданий базового уровня сложности (Б) – 1,
- для заданий повышенного уровня сложности (П) – 1.5,
- для заданий высокого уровня сложности (В) – 2.

D_i – вес вида проверяемой деятельности в задании:

- для воспроизведения представлений и знаний (1) – 1,
- для применения знаний и умений в стандартной ситуации (2) – 1.5,
- для применения знаний и умений в новой ситуации (3) – 2.

$$O_i = W_i * C_i * S_i * D_i$$

В табл. 8 приведены сведения о максимальных оценках, которые может получить участник экзамена за выполненное задание экзаменационной работы (табл. 5).

Задание	Первичный балл - W_i	Степень выполнения задания C_i (max)	Вес уровня сложности S_i	Вес вида проверяемой деятельности D_i	Максимальная оценка O_i
A1	1	1	1	1	1
A2	1	1	1.5	2	3
A3	1	1	1	1.5	1.5
A4	1	1	1	1.5	1.5
A5	1	1	1	1.5	1.5
A6	1	1	1.5	2	3
A7	1	1	1	1	1
A8	1	1	1.5	1.5	2.25
A9	1	1	1	1.5	1.5
A10	1	1	1.5	2	3
B1	1	1	1.5	2	3
B2	1	1	1	1.5	1.5
B3	1	1	1	1	1
B4	1	1	1.5	1.5	2.25
C1	2	1	2	1.5	6
C2	4	1	2	2	16
Итого по всем заданиям работы максимально:					49

Табл. 8. Оценки заданий по плану экзаменационной работы

2. Оценка степени выполнения олимпиадной работы в целом C_p производится следующим образом:

16

$$C_p = \sum_{i=1}^{16} C_i / 16.$$

Максимальная оценка степени выполнения работы – $C_p=1$.

3. Оценка за работу О с учетом «полуформата» ЕГЭ по информатике определяется как

16

$$O = 2 * \left(\sum_{i=1}^{16} O_i + C_p \right).$$

Максимальная оценка за всю работу О = 100 баллов.

Электронный и бумажный документооборот в ИС

В основу разработки инструментария ИС по документообороту заложены следующие принципы.

- В системе создается электронный отчет (протокол) каждой работы, который распечатывается организатором экзамена и после подписи участником отправляется на проверку.
- Задания части С распределяются между зарегистрированными в системе экспертами, протоколы экспертиз каждого задания в электронном виде хранятся в системе; бумажные протоколы экспертиз печатаются и подписываются экспертами.
- После шкалирования результатов в системе формируется отчет о результатах экзамена, печатается и предоставляется в предметную комиссию экзамена.

Программно-аппаратный комплекс системы тестирования

Программно-аппаратный комплекс системы для проведения ЕГЭ рассчитан на тысячу одновременных сессий участников и реализован в виде отказоустойчивого кластера с балансировкой нагрузки. Комплекс обеспечивает восстановление контекста сессии при сбое, шифрование данных при передаче и хранении, защиту от несанкционированного доступа на основе IP-адресов. Он состоит из двух связанных между собой территориально-распределенных центров обработки информации, показанных на рис. 8.

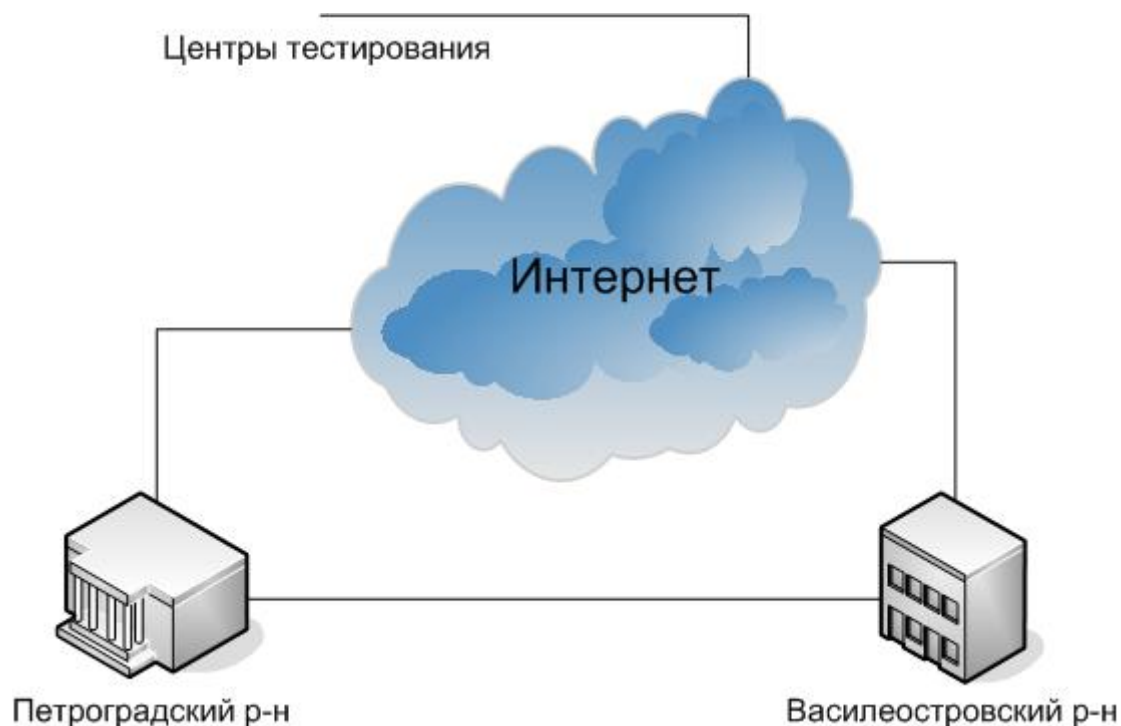


Рис. 8. Центры обработки информации

Каждый центр оснащен двумя серверами Sun Fire V490 и дисковым массивом Sun StorEdge 3310. Структура центра показана на рис. 9. На один центр приходится восемь двухъядерных процессоров UltraSPARC IV+ и 32 ГБ оперативной памяти. На каждом сервере установлено следующее программное обеспечение:

- операционная система Solaris 10;
- СУБД Oracle 10g;
- система дистанционного обучения AcademicNT;

- Web-сервер Tomcat 6;
- Web-сервер Apache 2;
- среда выполнения Java 6.

Наличие двух Web-серверов обусловлено распределением ролей. Tomcat обеспечивает работу Java-сервлета системы дистанционного обучения, а Apache – балансировку нагрузки на экземплярах сервера Tomcat и шифрование данных. Сбой в каком-либо из перечисленных приложений, выход из строя сервера или всего центра не препятствует проведению ЕГЭ и не приводит к потере данных.

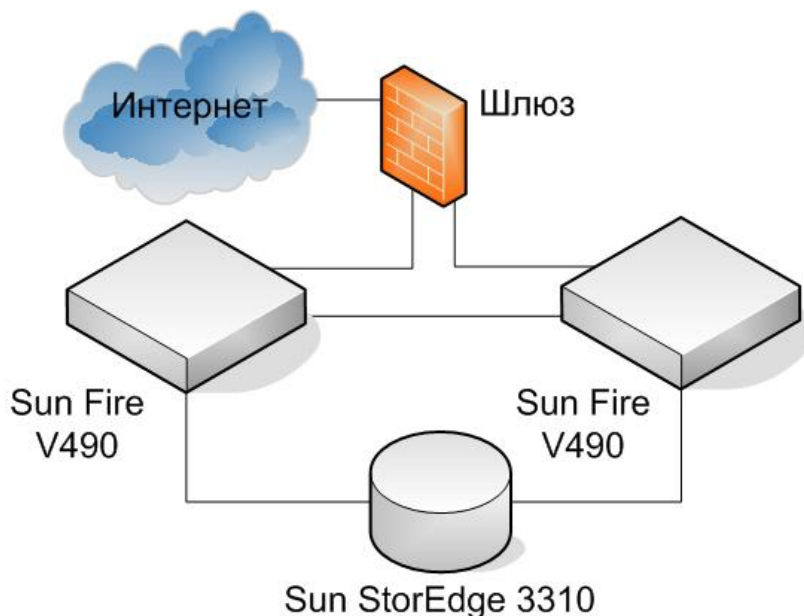


Рис. 9. Структура центра обработки информации

Результаты апробации системы тестирования в олимпиаде СПбГУ ИТМО 25 февраля 2007 г.

Для участия в олимпиаде были приглашены школьники выпускных классов Санкт-Петербурга. После входного тестирования было отобрано для участия 137 чел. Фактически в олимпиаде участвовало 133 чел. (рис. 10).



Рис. 10. Явка участников олимпиады

В табл. 9 приведены сведения о результатах выполнения заданий базового уровня сложности (ожидаемый результат для ЕГЭ – 60–80%), а в табл. 10 – заданий повышенного уровня сложности (ожидаемый результат для ЕГЭ – 40–60 %). Задания высокого уровня сложности относятся к части С (ожидаемый результат для ЕГЭ – 10–30%). На рис. 11 приведены сведения о результатах выполнения заданий части С, а на рис. 12–13 – сведения о распределении первичных баллов у участников олимпиады, выполнивших задания С1 и С2 соответственно. Количество работ с автоматическим установлением работоспособности программ задания С1 и оцененных системой в 2 балла – 4 из 63, что составляет 6,35%. На рис. 14 приведены сведения о результатах олимпиадной работы в первичных баллах (средний первичный балл за работу – 11,35), а на рис. 15 – в шкалированных баллах (средний балл за работу в 100-бальной шкале – 43,95). На рис. 16 приведены сведения о распределении дипломов между участниками олимпиады (средняя оценка за работу – 3,58), а в табл. 11 – распределение шкалированных баллов между разделами информатики.

№	Обозначение задания	Содержание задания	% выполнения задания
1	A1	Кодирование текстовой информации. Кодировка ASCII. Основные используемые кодировки кириллицы	86,5
2	A3	Умения выполнять арифметические операции в двоичной, восьмеричной и шестнадцатеричной системах счисления	82
3	A4	Знание и умение использовать основные алгоритмические конструкции: следование, ветвление, цикл	99,2
4	A5	Использование переменных. Объявление переменной (тип, имя, значение). Локальные и глобальные переменные	95
5	A7	Умение представлять и считывать данные в разных типах информационных моделей (схемы, карты, таблицы, графики и формулы)	78
6	A9	Формальное исполнение алгоритма, записанного на естественном языке	97
7	B2	Умение исполнять алгоритм в среде формального исполнителя	76
8	B3	Знания о файловой системе организации данных, о технологиях создания и обработки информации в электронных таблицах, базах данных и графических редакторах	89
Средний процент выполнения заданий базового уровня сложности:			87,8

Табл. 9. Результаты выполнения заданий базового уровня сложности

№	Обозначение задания	Содержание задания	% выполнения задания
1	A2	Умение подсчитывать информационный объем сообщения	48
2	A6	Знание основных понятий и законов математической логики	88
3	A8	Умение строить и преобразовывать логические выражения	69
4	A10	Умение определять скорость передачи информации при заданной пропускной способности канала	60
5	B1	Умение прочесть фрагменты программы на языке программирования и найти допущенные ошибки	12
6	B4	Умения строить таблицы истинности и логические схемы	68
Средний процент выполнения заданий повышенного уровня сложности:			57,5

Табл. 10. Результаты выполнения заданий повышенного уровня сложности

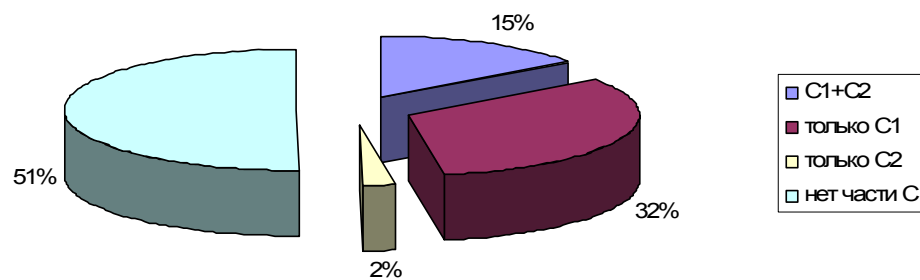


Рис. 11. Результаты выполнения части C

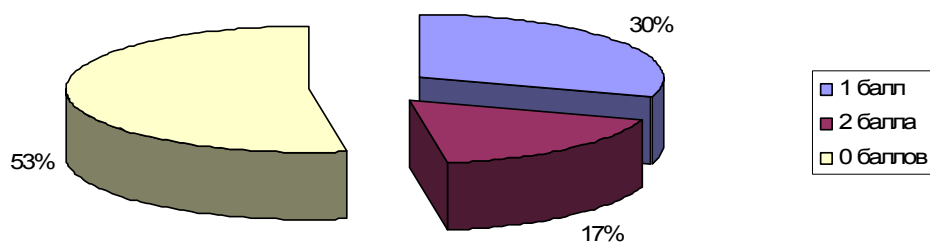


Рис. 12. Распределение баллов в C1

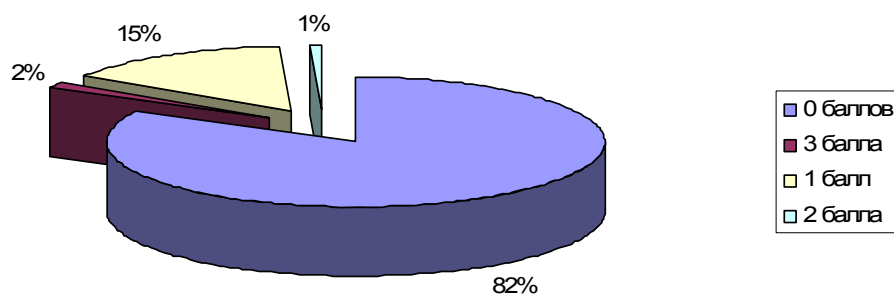


Рис. 13. Распределение баллов в C2

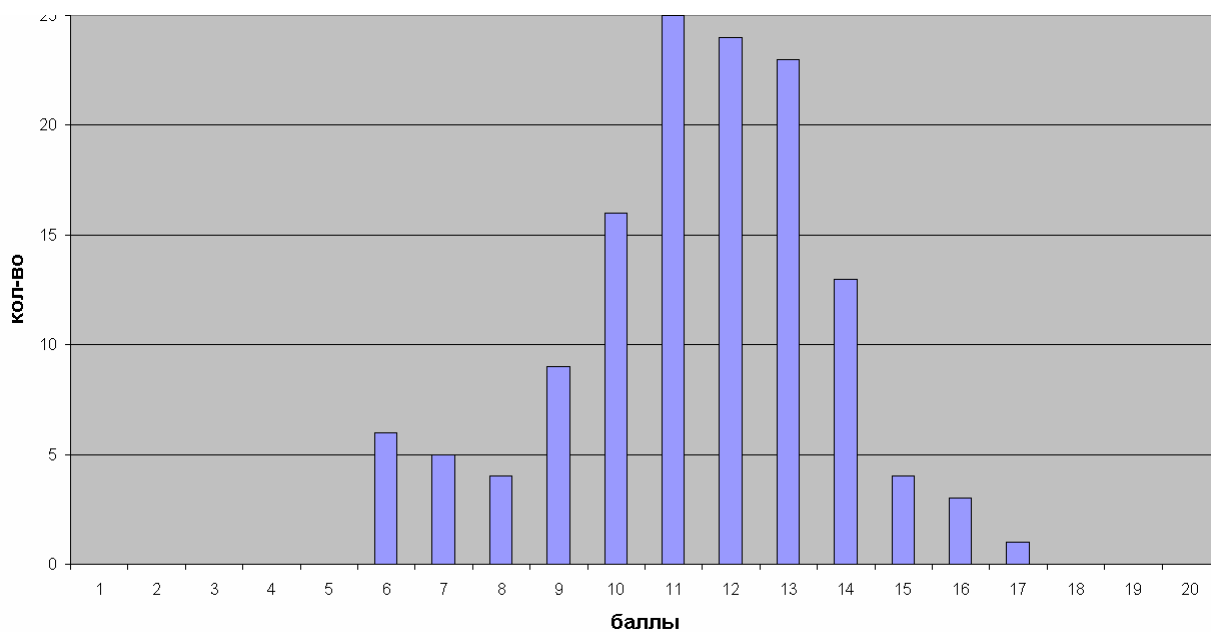


Рис. 14. Результаты олимпиады по информатике 25.02.07 в первичных баллах

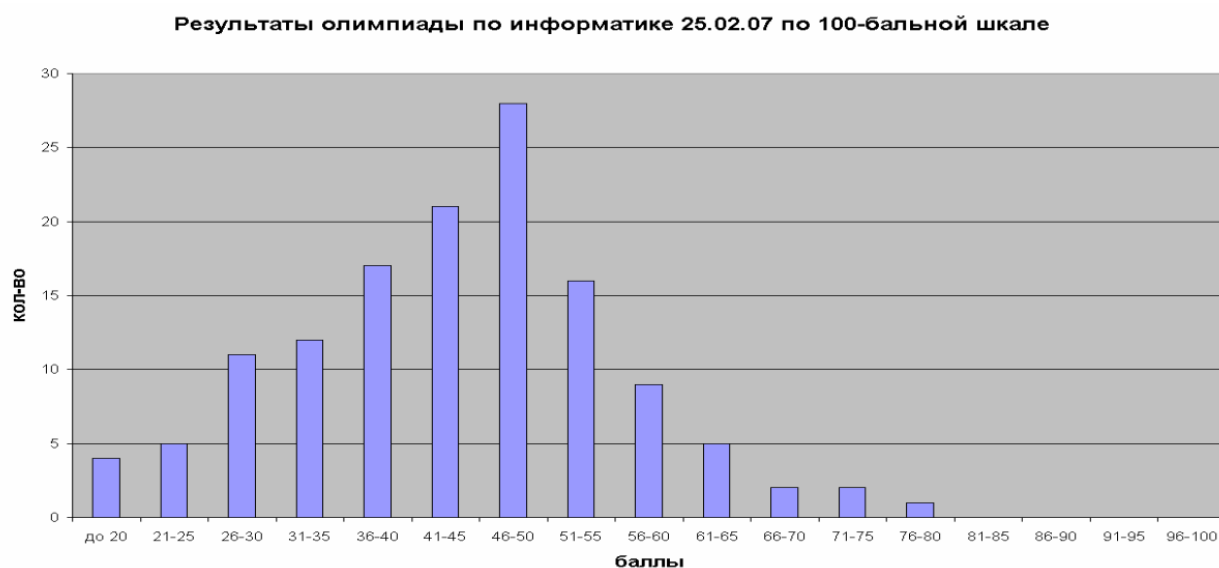


Рис. 15. Результаты олимпиады по информатике 25.02.07 по 100-балльной шкале

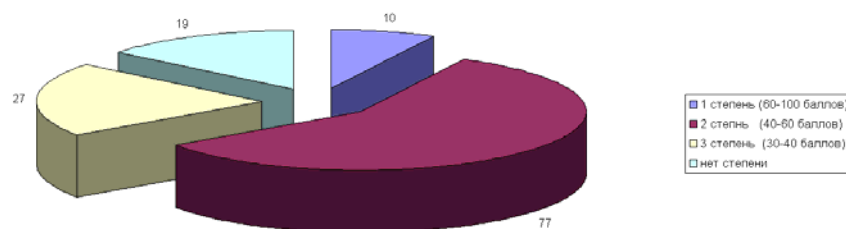


Рис. 16. Распределение дипломов олимпиады по информатике по степеням

№ п/п	Название раздела	Доля раздела в оценке		
		В виде первичных баллов		В 100- бальной шкале олим- пиады
		ЕГЭ	олимпиада	
1.	Информация и ее кодирование	17.5	20	17
2.	Алгоритмизация и программирование	32.5	30	25
3.	Основы логики	12.5	15	15
4.	Моделирование и компьютерный эксперимент	2.5	5	2
5.	Программные средства и технологии ИКТ	20	5	2
6.	Технология программирования	15	25	39

Табл. 11. Распределение баллов между разделами курса «Информатика»

Выводы и рекомендации по итогам апробации

1. Задания базового уровня А1, А2, А4, А5, А9, В3, а также задания повышенного уровня А6, А8, В4 следует переработать для проведения городской олимпиады Санкт-Петербурга по информатике в компьютерном виде в рамках запланированного содержания для олимпиадного уровня.

2. Задание В1 (повышенный уровень, проверка знаний и умений в стандартной ситуации) является самым неуспешным в олимпиадной работе (всего 12 % правильных ответов при 40–60% ожидаемых). Предметной комиссии следует разработать новые КИМы в рамках запланированного содержания задания В1.
3. Задания части С являются заданиями олимпиадного уровня. Предметной комиссии рекомендуется разработка аналогичных заданий для городской олимпиады г. Санкт-Петербурга по информатике в компьютерном виде.
4. В результате шкалирования результатов олимпиады установлено соотношение между основными блоками курса информатики «Алгоритмизация и технологии программирования»: «Основы логики и теории кодирования информации» как 2:1.

Литература

1. Васильев В.Н., Лисицына Л.С., Мартыненко А.В. Аналитический отчет предметной комиссии Санкт-Петербурга ЕГЭ по информатике в 2006 году. СПб: ЦПИ, 2006. – 26с.

СЕТЕВОЕ СООБЩЕСТВО МЕТОДИЧЕСКОГО ИНТЕРНЕТ-ЦЕНТРА: СОВРЕМЕННОЕ СОСТОЯНИЕ И ПЕРСПЕКТИВЫ РАЗВИТИЯ

Л.С. Лисицына

В статье рассмотрены вопросы, связанные с развитием сетевого сообщества МИЦ в 2006 году. Представлена структура сетевого сообщества, приведены сведения о развитии инициатив, связанных с разработкой ДОТ для подготовки преподавателей-тьюторов, создания «методической копилки» МИЦ в виде аннотированного каталога, проведения Интернет-семинаров в сетевом сообществе МИЦ.

Введение

Развитие методического Интернет-центра (МИЦ) <<http://ito-center.ifmo.ru>> [1] в 2006 г. проходило в рамках выполнения научно-исследовательской работы с Национальным фондом подготовки кадров № ELSP/B1/Gr/004/020-06/360030 «Интернет-поддержка профессионального развития педагогов» (2006–2008 гг.).

Взаимодействие виртуальных центров в МИЦ было реализовано и апробировано в сетевом сообществе МИЦ, которое включало в 2005 г. районные научно-методические центры (НМИЦ) и другие ОУ системы образования Санкт-Петербурга [2]. За период с мая 2005 по декабрь 2006 г. произошло расширение сетевого сообщества МИЦ за счет государственных университетов и образовательные учреждения (ОУ) системы дополнительного профессионального образования (ДПО) России, ведущих практическую деятельность по повышению ИКТ-компетентности работников образования и пользующихся методической поддержкой в МИЦ. За это время наблюдался не только количественный рост виртуальных представительств (к концу 2006 г. их было 65) и пользователей системы (к концу 2006 г. их было 258 чел.), но и совершенствование технологий и инструментов МИЦ для реализации новых инициатив:

- по разработке дистанционных образовательных технологий (ДОТ) для подготовки методистов и преподавателей-тьюторов системы ДПО работников образования (в роли преподавателя-тьютора в МИЦ в 2006 г. было зарегистрировано 277 чел., а успешно прошло подготовку 240 чел);
- по проведению Интернет-семинаров и Интернет-конференций по актуальной тематике (в 2006 г. в МИЦ прошло три Интернет-семинара, на форумах которых было 96 публикаций с оригинальными авторскими учебно-методическими материалами (УММ));
- по созданию «методической копилки» в виде аннотированного каталога учебно-методических ресурсов, используемого центром разработки (ЦР) для доработки образовательного пространства МИЦ (только за период с ноября 2006 г. по апрель 2007 г. в каталоге появилось более 1000 аннотированных ресурсов).

Интернет-семинары для сетевого сообщества МИЦ

Структура сетевого сообщества к концу 2006 г. представлена на рис. 1. Характерной особенностью периода с 2005г. по 2006 г. является усиление мотивации к сетевому общению не только с конкретными разработчиками базовых образовательных модулей БОМ [3], но и в сообществе. Только во второй половине 2006 г. было проведено три Интернет-семинара, которые явились откликом на публикацию ЦР МИЦ популярных образовательных программ «Основы ИКТ для преподавателя-предметника» и «ИКТ для администратора ОУ» вместе с их электронными УММ. В результате сетевого взаимодействия на страницах форумов Интернет-семинаров, посвященных обсуждению методик данных программ, появились публикации участников семинаров с аль-

тернативными (авторскими) УММ. В этой связи появилась идея сбора, анализа и отбора новых методических материалов для последующей доработки образовательных модулей МИЦ. На рис. 2. приведены сведения о распределении публикаций в Интернет-семинарах, которые проводились в МИЦ в 2006 г. (всего было собрано от участников Интернет-семинаров 96 публикаций с оригинальными авторскими УММ).

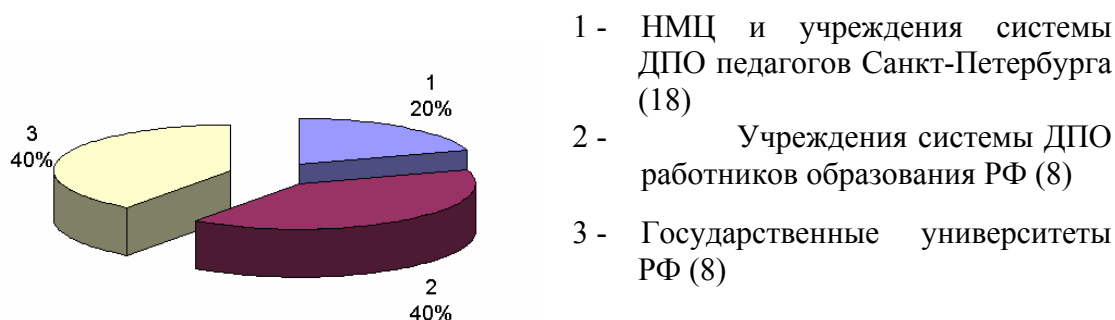


Рис. 1. Структура сетевого сообщества МИЦ в 2006 г.

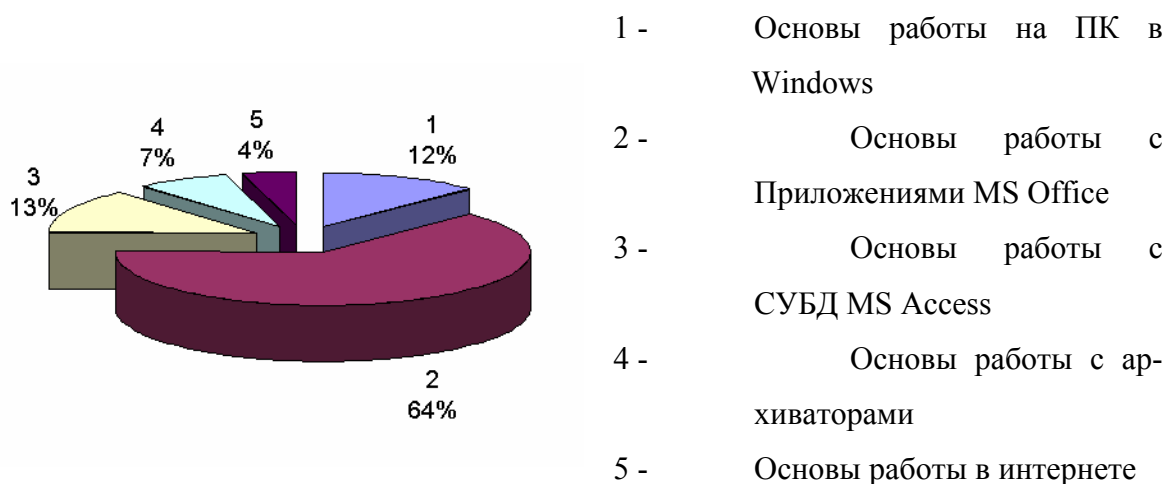


Рис. 2. УММ участников Интернет-семинаров МИЦ в 2006 г.

Разработка ДОТ для подготовки в МИЦ преподавателей-тьюторов

Главная цель подготовки преподавателей-тьюторов – распространение передового педагогического опыта, накопленного в профессиональном сообществе МИЦ, и его внедрение в практику компетентностного повышения квалификации работников образования в области ИКТ. Вместе с этим, подготовка преподавателей-тьюторов в МИЦ способствует распространению идей модульного компетентностного подхода в сфере ДПО, внедрению новой культуры сетевого общения между преподавателями и дистанционного обучения слушателей на основе цифровых образовательных ресурсов (ЦОР).

Тьюторская программа – это образовательная программа, построенная на основе изучения одного из модульных учебных курсов ЦР и реализуемая в МИЦ на основе ДОТ. *Преподаватель-тьютор* (далее *тьютор*) – это педагог, хорошо знающий предметную область и имеющий ИКТ-компетентность не ниже базового уровня по каждой из осваиваемых в тьюторской программе компетенций. *Преподаватель-консультант* – это преподаватель ЦР, как правило, являющийся одним из разработчиков изучаемого модульного учебного курса и ведущий подготовку тьюторов.

Основными задачами тьюторской программы являются:

- углубленное освоение компетенций, входящих в изучаемый модульный учебный курс;
- освоение компетенций, связанных с методиками дистанционного преподавания изучаемого модульного учебного курса;
- освоение компетенций, связанных с изучением и использованием эффективных приемов сетевого общения тьюторов с преподавателями-консультантами и между собой;
- стажировка тьютора на курсах повышения квалификации работников образования по программе изучаемого модульного учебного курса.

Дистанционное обучение и консультирование тьюторов в сетевой информационно-образовательной среде МИЦ направлено на освоение двух групп компетенций:

1. предметных компетенций, полученных при детализации исходной компетенции РО «Применять ИКТ в образовательной деятельности»;
2. компетенций по формированию следующих навыков дистанционного взаимодействия:
 - *педагогического субъект-субъектного* взаимодействия с преподавателем-консультантом и освоение дистанционных технологий проведения занятий, консультаций, аттестаций;
 - *организационного субъект-субъектного* взаимодействия с разработчиками образовательных модулей и курсов, с администрацией МИЦ, с другими преподавателями-тьюторами;
 - *педагогического субъект-ресурсного* взаимодействия в МИЦ с сетевыми образовательными и аттестующими ЦОР;
 - *организационного субъект-ресурсного* взаимодействия с сетевыми ресурсами для планирования и организации учебного процесса, с электронным документооборотом дистанционного обучения.

При разработке ДОТ для подготовки преподавателей-тьюторов следует руководствоваться следующими принципами:

- каждая элементарная компетенция (ЭК) [3], входящая в перечень изучаемого модульного учебного курса, осваивается тьютором на углубленном уровне компетентности (принцип углубленного обучения тьютора);
- освоение каждой ЭК контролируется в виде примеров и заданий, которые тьютор разрабатывает для проведения практических занятий по этой теме курса и публикует на специальном форуме в МИЦ для проверки и рецензирования преподавателем-консультантом (промежуточная аттестация тьютора по освоению предметных компетенций и компетенций педагогического субъект-субъектного и субъект-ресурсного взаимодействия);
- целевая компетентность изучаемого модульного учебного курса контролируется в виде плана подготовки выпускной работы, который тьютор разрабатывает для слушателей этого курса и публикует на специальном форуме в МИЦ для проверки и рецензирования преподавателем-консультантом (заключительная аттестация тьютора по освоению предметных компетенций и компетенций педагогического субъект-субъектного и субъект-ресурсного взаимодействия);
- итоговая аттестация тьютора заключается в подготовке под руководством преподавателя-консультанта пробной лекции или практического занятия и их апробации на курсах повышения квалификации слушателей (педагогическая стажировка тьютора);
- методические материалы, подготовленные и апробированные тьютором на курсах повышения квалификации слушателей, публикуются тьюторами на специальном

форуме МИЦ и обсуждаются совместно с другими преподавателями-тьюторами и с преподавателями-консультантами в рамках проведения Интернет-семинаров (заключительная аттестация тьютора по освоению компетенций педагогического и организационного субъект-субъектного взаимодействия).

В 2006 г. было подготовлено 103 преподавателя для системы ДПО педагогов Санкт-Петербурга, на рис. 3 приведены сведения о распределении выпускников курсов.

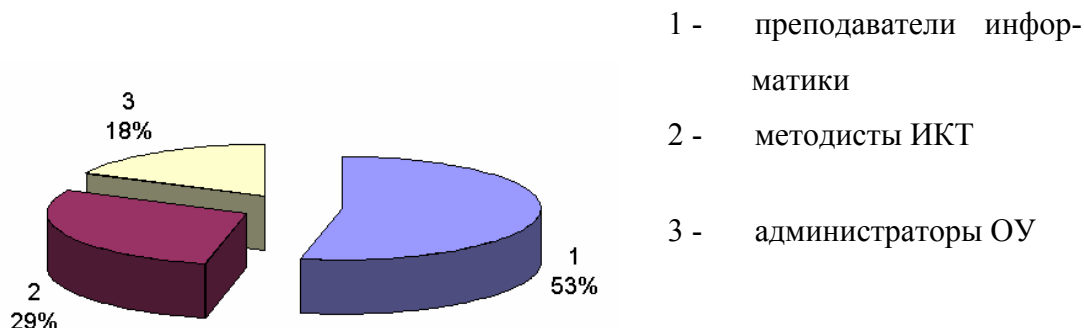


Рис. 3. Распределение выпускников курсов по подготовке преподавателей-тьюторов в МИЦ в 2006 г.

Создание аннотированного каталога ресурсов МИЦ

Для работы с каталогом в МИЦ введена роль «Редактор каталога»; разработана сетевая среда для внесения, модификации и удаления ресурса из каталога МИЦ; разработана сетевая среда для открытого поиска и просмотра ресурсов в каталоге.

«Каталог» – это раздел МИЦ, доступ к которому имеют пользователи с ролью «Администратор» или «Редактор каталога» для дистанционных публикаций новых учебно-методических материалов (УММ) или ссылок на них. В этом разделе пользователь может добавлять в хранилище МИЦ, редактировать и удалять ресурсы каталога, владельцем которых он является, а также осуществлять поиск по каталогу.

Выбор в меню рубрики «Каталог» приведет к окну просмотра и поиска списка ресурсов аннотированного каталога. Для создания новой записи необходимо воспользоваться кнопкой «Добавить» и в открывшейся форме заполнить следующие поля:

- название;
- URL;
- аннотация.

Для создания аннотированного каталога УММ сообщества для доработки образовательного пространства МИЦ была доработана сетевая среда МИЦ, и ряд виртуальных центров МИЦ получил право заполнять аннотированный каталог ресурсов МИЦ. В каталоге осуществляется сбор по темам, типам и форматам ресурсов. Материалы каталога аккумулируют лучшие материалы по итогам рейтингового голосования среди участников Интернет-семинаров и методических форумов, по итогам конкурсов на создание лучшего ЦОР по данной тематике и т.п. Аннотированный каталог является «методической копилкой» сообщества МИЦ, своеобразным местом публичной экспертизы и апробации новых УММ у коллег-единомышленников.

Литература

1. Васильев В.Н., Лисицына Л.С., Лямин А.В. Методический Интернет-центр. СПб: Питер, 2005. 96 с.
2. Васильев В.Н., Лисицына Л.С. От курсов повышения квалификации на базе СПбГУ ИТМО – к созданию методического Интернет-центра для учреждений системы дополнительного профессионального педагогического образования Санкт-Петербурга // Сб. «Развитие информатизации образования Санкт-Петербурга (состояние, опыт, перспективы)». СПб: ЦПИ, 2005. С. 47–54.
3. Лисицына Л.С. Теория и практика компетентностного обучения и аттестаций на основе сетевых информационных систем. СПб: СПбГУ ИТМО, 2006. 147 с.

СИСТЕМА ДИСТАНЦИОННОГО ОБУЧЕНИЯ ВТОРОГО ПОКОЛЕНИЯ

В.Н. Васильев, А.В. Лямин, М.С. Чежин

Статья посвящена развитию дистанционных компьютерных технологий обучения в СПбГУ ИТМО. Рассмотрена история развития системы дистанционного обучения университета, описаны организация и возможности современной системы AcademicNT. Приведены основные результаты использования системы в учебном процессе и определены основные проблемы, которые необходимо решить для повышения эффективности ее использования.

Введение

Одним из приоритетных направлений развития высшей школы является информатизация образования, осуществляемая путем внедрения в учебный процесс современных методов и технологий дистанционного обучения, позволяющих повысить качество, доступность и непрерывность образования и тем самым способствовать созданию открытого образовательного пространства в форме системы открытого образования.

Современное состояние информатизации образования характеризуется опережающим развитием в образовательных учреждениях технической оснащенности и информационных технологий по сравнению с процессом создания новых педагогических методов и технологий. Это побуждает вузы к самостоятельной интенсивной работе по совершенствованию учебно-методического обеспечения современных компьютерных технологий обучения и создания систем дистанционного обучения.

Развитие системы дистанционного обучения СПбГУ ИТМО

Дистанционное обучение (ДО) – это комплекс образовательных услуг, предоставляемых различным категориям граждан с использованием современных информационных и телекоммуникационных технологий при территориальной разобщенности преподавателя и студентов. Эта технология обучения реализуется в среде специализированной информационно-образовательной системы, которая называется системой дистанционного обучения и представляет собой системно организованную совокупность информационных ресурсов, средств передачи данных, аппаратно-программного и организационно-методического обеспечения, ориентированную на удовлетворение образовательных потребностей пользователей. Неотъемлемой частью системы ДО является сетевая база данных, в которой хранятся информационные и образовательные ресурсы системы, информация о правах и уровнях доступа всех пользователей системы, а также информация о результатах контроля знаний обучающихся в системе.

Задачи системы ДО – активизация самостоятельной работы студентов, индивидуализация и интенсификация обучения; унификация и объективизация контроля усвоения уровня знания дисциплины; расширение спектра образовательных услуг вуза; развитие возможностей сохранения единства образовательного пространства; обеспечение широкого доступа к информационно-образовательным ресурсам вуза; экспорт образовательных услуг на внешний образовательный рынок.

СПбГУ ИТМО включился в создание средств дистанционного обучения почти 10 лет назад, в марте 1998 г., когда Ученый совет открыл целевую комплексную программу развития системы дистанционного обучения [1, 2]. Для реализации этой программы в 1999 г. в Университете был создан центр дистанционного обучения (ЦДО). На ЦДО возлагались задачи организационно-методического и программно-технического сопровождения дистанционного обучения, совершенствование и развитие системы ДО, обеспечение учебного процесса в системе ДО. Создание и разработка этой комплексной программы базировалось на опыте уже разработанных и функционирующих систем

компьютерного обучения и тестирования на ряде кафедр Университета по естественно-научным и гуманитарным дисциплинам, вычислительной технике и по целому ряду других технических дисциплин.

Было принято решение строить систему ДО в СПбГУ ИТМО на основе сетевой Интернет-технологии, которая позволяет осуществлять доступ к образовательным и информационным ресурсам системы с любого подключенного к сети Интернет или к любой другой IP-сети персонального компьютера посредством программы-клиента Web-навигатора (обозревателя) [3–5]. Сетевая технология для СПбГУ ИТМО является экономически наиболее целесообразной и перспективной, в отличие от CASE- и локальной мультимедиа-технологии, тем более что к этому моменту времени в Университете сформировалась и развитая телекоммуникационная инфраструктура, имеющая выход в глобальные компьютерные сети (RUNNet, Internet). Таким образом, в Университете были все необходимые условия для перехода на новый уровень развития информационных технологий обучения.

Основной принцип построения системы заключался в том, что она должна быть не просто совокупностью различных курсов ДО, разработанных специалистами по компьютерным технологиям, а должна представлять собой открытый, инвариантный к содержанию инструментарий-оболочку, позволяющую любому преподавателю, не владеющему программированием и компьютерными технологиями, создавать и использовать курсы дистанционного обучения в соответствии со своей научно-педагогической квалификацией и опытом преподавания.

На первом этапе были разработаны такие модули системы ДО, как автоматизированная система для проведения аттестующего и обучающего тестирования и электронная библиотека, обеспечивающая возможность удаленного доступа к учебным материалам. Функционирование системы ДО осуществлялось на основе эффективных стандартных Интернет-технологий:

- HTML-обозреватель (Netscape Navigator, Internet Explorer или любой другой Java-совместимый обозреватель) обеспечивает гипертекстовый интерфейс взаимодействия пользователя с системой;
- Java-приложения обеспечивают интерактивность и взаимодействие с базой данных;
- Web-сервер передает HTML-документы и приложения обучаемым;
- сервер базы данных на основе СУБД Oracle позволяет хранить всю информацию, используемую при обучении, включая информационно-обучающие элементы системы, результаты аттестации знаний, информацию о пользователях и их правах.

Внедрение системы ДО в учебный процесс показало правильность принятых решений и перспективность данного направления. Автоматизация процесса обучения позволила студентам контролировать свои знания в процессе изучения дисциплины, а преподавателю дало возможность оценить структуру и качество подготовки учебного материала.

Наряду с программно-техническим и учебно-методическим формировалось нормативно-организационное и кадровое обеспечения системы ДО. Основу нормативного обеспечения учебного процесса в системе ДО составили:

- положение о целевой научно-методической комплексной программе «Развитие системы дистанционного обучения в ИТМО»;
- положение об экспертизе элементов компьютерного обучения системы ДО;
- положение о признании элементов компьютерного обучения системы ДО в качестве учебно-методического труда;
- положение о проведении текущего контроля успеваемости и промежуточной аттестации студентов в системе дистанционного обучения.

Все более широкое использование системы ДО в учебном процессе Университета потребовало расширения технических возможностей ЦДО, и с октября 2003 г. в экс-

плутацию был введен учебный класс, целиком оснащенный рабочими местами и серверами производства компании Sun Microsystems и рассчитанный на одновременное обслуживание шестидесяти пользователей. Использование этого класса со специально разработанной системой формирования индивидуального расписания студентов позволили обеспечить массовый текущий контроль успеваемости в компьютерной форме. Уже в осеннем семестре 2003/2004 учебного года с системой тестирования работал 3851 студент, было проведено 53525 сеансов аттестаций, во время которых студентам было предложено ответить на 750476 вопросов. Через год, в осеннем семестре 2004/2005 учебного года, в системе проходили аттестацию 4309 студентов, проведено 77137 сеансов аттестаций, что в 1,44 раза больше, чем за такой же период прошлого года. Большинство аттестаций пройдено из компьютерного класса ЦДО, вместе с тем в режиме обучения система активно использовалась и с внешних IP-адресов, в результате чего среднее время, проведенное студентами в системе, составило 17 академических часов.

Опыт использования системы показывал необходимость ее существенного обновления. Необходимо было применить комплексный подход к организации учебно-методических материалов, обеспечить широкие коммуникативные возможности участников учебного процесса, усовершенствовать технологии администрирования. Другими словами, подойти к системе ДО как к информационной системе для проведения учебного процесса. В результате была разработана система ДО второго поколения – AcademicNT, которая с 2006 г. успешно используется в учебном процессе СПбГУ ИТМО. В настоящий момент в системе дистанционного обучения представлено 120 программ обучения, в том числе:

- общепрофессиональных дисциплин – 43;
- гуманитарных и социально-экономических дисциплин – 18;
- математических и естественнонаучных дисциплин – 39;
- специальных дисциплин и дисциплин специализации – 20.

Организация и возможности системы ДО AcademicNT

В системе AcademicNT учебно-методические материалы в совокупности образуют электронный учебно-методический комплекс (УМК), который имеет иерархический, модульный характер [6]. Верхнюю ступень в структуре УМК занимает учебный план подготовки учащегося по определенной специальности. Рабочие программы дисциплин, определенных учебным планом, составляют второй уровень. Каждая дисциплина может включать несколько электронных курсов. Каждый электронный курс, в свою очередь, состоит из ряда аттестующих, обучающих и информационных модулей. В системе ДО AcademicNT предлагается следующий набор программно-технических модулей, позволяющий реализовывать все виды учебной работы:

- электронные конспекты;
- информационные ресурсы;
- электронные тесты;
- виртуальные лабораторные работы;
- электронные практикумы.

Модульный принцип построения электронного курса предоставляет широкие возможности авторам УМК и делает систему ДО более гибкой в использовании, так как отдельные виды учебной работы могут обеспечиваться блоками из нескольких модулей, что даст наиболее полное соответствие авторскому замыслу. Количество элементов электронных учебно-методических материалов и их объем должны соответствовать видам учебной работы и их трудоемкости, предусмотренным рабочей программой дисциплины, и должны быть разработаны на основе государственных образовательных

стандартов. Учащийся, использующий УМК, видит все элементы во взаимосвязи, что существенно облегчает его работу.

Программа дисциплины построена на основе стандартной формы и включает разделы: цели и задачи дисциплины, требования к уровню освоения содержания дисциплины, объем дисциплины в часах, виды контроля по семестрам, структура дисциплины, учебно-методическое, программное и материально-техническое обеспечение дисциплины, инструкции. Программа должна содержать методические рекомендации преподавателям и методические указания для студентов по всем видам учебной работы и формам занятий по дисциплине, в том числе курсовым работам и проектам. В разделе «Учебно-методическое обеспечение дисциплины» список основной и дополнительной литературы, используемой при изучении дисциплины, обеспечивает автоматизированный контроль книгообеспеченности учебного процесса Университета.

Одним из основных программно-технических модулей электронного курса является электронный конспект – модуль, основанный на структурированной и разделенной на уровни совокупности информационных страниц и кадров. Содержание информационных страниц отображается на экране в гипертекстовом и графическом виде (схемы, рисунки, графики, сложные формулы, фотографии). Обеспечена возможность использования на страницах электронного конспекта элементов мультимедиа (анимация, видео, аудио). Например, на страницах конспекта могут размещаться демонстрации на основе компьютерного моделирования физических процессов, позволяющие проводить эксперименты, получать различных характеристик элементов и устройств, при этом студент может менять параметры изучаемого устройства.

Структура электронного конспекта может быть построена с использованием принципа многоуровневости (многослойности) изложения учебного материала. В этом случае материал, изложенный на страницах электронного конспекта, структурируется не только в виде линейной структуры, как это принято в печатных изданиях (например, раздел, подраздел, глава, параграф и т.п.), но и по набору уровней (слоев) изложения (например, абстракт, конспект, полный текст, примеры, упражнения, задачи и т.п.). В результате каждая структурная единица имеет несколько уровней изложения, что при построении электронного конспекта позволяет автору реализовать хорошо структурированное, многослойное или многоуровневое изложение материала.

Окно электронного конспекта имеет панель навигации, содержащую кнопки перехода вперед и назад по страницам конспекта, поле для выбора уровня изложения материала, элементы управления для поиска страниц и кадров, оглавление конспекта. Управление навигацией в электронном конспекте задается на уровне описания структуры конспекта, и автору материала нет необходимости создавать специальный интерфейс, позволяющий переходить от одной страницы конспекта к другой. В электронных конспектах поддерживается как полнотекстовый поиск необходимой информации, так и поиск по отдельным элементам – названиям, ключевым словам. Электронный конспект должен содержать информационные кадры с небольшим объемом информации вспомогательного, справочного характера по отношению к материалу страниц электронного конспекта. В качестве такой информации могут выступать определения, иллюстрации, Flash-анимации и т.д. Доступ к информации, размещенной в кадре, учащийся получает по ссылке с информационной страницы конспекта, при этом кадр не входит в основную структуру электронного конспекта. Кроме этого, обращение к информации, размещенной в информационных кадрах, возможно и с помощью процедуры поиска, и путем выбора нужного кадра из списка. В результате с помощью набора информационных кадров в конспекте может быть сформирован глоссарий или другие информационные структуры (например, персоналии, словари, справочники), строящиеся на принципе «ключевое слово–содержание». При необходимости формирования в разделе электронного курса глоссария или справочника с большим объемом информа-

ции на каждой странице его можно сделать в виде отдельного конспекта. Подобный конспект может быть видимым или невидимым. В первом случае конспект будет присутствовать в явном виде в качестве самостоятельного модуля в структуре курса и будет доступен для отдельного изучения. Во втором случае конспект в структуре курса виден не будет, и обращение к страницам этого конспекта будет осуществляться лишь по ссылкам со страниц других конспектов. Таким образом, в разделе электронного курса возможно размещение нескольких конспектов – конспект с основным содержанием раздела, энциклопедия, глоссарий, задачник с контрольными работами и примерами решений, описание лабораторных работ, справочная информация.

Программно-технический модуль «Электронный конспект» является обучающим модулем электронного курса, предназначенным для организации самостоятельной работы студентов при изучении ими теоретических положений изучаемой дисциплины. С помощью электронного конспекта возможна реализация учебно-методических модулей различного типа. Использование материалов электронного конспекта пользователями осуществляется при работе в сети, т.е. в режиме on-line.

Для размещения в системе ДО AcademicNT учебно-методических материалов, предназначенных для скачивания и последующей работы с ними в режиме off-line, используется модуль «Информационный ресурс». В качестве информационного ресурса могут выступать презентации к лекциям, чертежи, рисунки, тексты программ, базы данных, т.е. любая информация, организованная в виде файла. Никаких ограничений на тип файла не накладывается. При необходимости использовать группу файлов они архивируются в ZIP-архив. Информационные ресурсы являются вспомогательными материалами, предназначенными для выполнения самых различных заданий и работ по данному учебному курсу, например в форме информационного ресурса в УМК могут быть размещены элементы методического блока – презентации лекций, плакаты.

Обязательным элементом каждого раздела электронного курса являются электронные тесты. Электронные тесты являются эффективным средством определения уровня знаний студентов. Во время тестирования студенту последовательно предъявляются кадры. В системе определено более 40 разновидностей кадров, в которых реализованы различные интерфейсы. К базовой группе тест-кадров относятся: информационный кадр, задание закрытого типа, задание открытого типа, задание на установление правильной последовательности и задание на установление соответствия. Кроме этого, определены группы тестовых заданий графического и бланкового типов. В тестовых заданиях графического типа основой вопроса и объектом для ответа является рисунок. В зависимости от параметров и способа формирования ответа определены графические задания закрытого типа с одним и несколькими правильными ответами, открытого типа с одним и с несколькими ответами, на установление последовательности и задание одной или нескольких связей, на задание маршрута и на соответствие. Вопросы бланкового типа представляют собой сложные, комбинированные вопросы, состоящие из нескольких элементов, и могут включать поля ввода, списки, ячейки, возможности выделения и перемещения элементов. Последовательность кадров формируется системой на основе алгоритма определенного разработчиком теста. Это может быть и псевдослучайный алгоритм, и жестко определенная последовательность, и алгоритм, когда при выборе следующего кадра учитывается ответ обучаемого на предыдущий.

Электронные тесты в системе ДО AcademicNT подразделяются на обучающие и аттестующие. Обучающий режим теста предназначен прежде всего для самоконтроля студента и определения траектории обучения: в зависимости от ответов студента ему будут предъявляться те или иные обучающие элементы. В обучающем режиме особое внимание должно быть уделено формированию диалога системы и пользователя путем задания вариантов реакции системы на возможные различные действия студента при

прохождении теста. В результате обеспечивается высокая степень интерактивности модулей. Система предоставляет учащемуся возможности активного взаимодействия с модулем, при котором реализуется обучающий диалог с целью выработки у обучаемого наиболее полного и адекватного знания сущности изучаемого материала. Основными компонентами обучающего диалога можно считать:

- обучающее воздействие, соответствующий ему обучающий модуль УМК или его кадр;
- контроль (проверка) исполнения данного учебного фрагмента;
- корректировка обучающего воздействия в соответствии с результатами контроля, т.е. обратная связь.

Наличие обучающего диалога (интерактива) обеспечивает эффект присутствия преподавателя, когда каждый обучаемый по любому вопросу, при любой ошибке, например, при решении задач, получает необходимый именно ему учебный материал. В результате при изучении электронного УМК системой ДО формируется индивидуальная траектория обучения для каждого студента, т.е. система выполняет роль электронного тьютора. В процессе изучения материала системой могут быть обеспечены многочисленные объектно-ориентированные подсказки, «проявляющиеся» по мере необходимости. В результате реализуется уровень интерактивности «реального масштаба времени», при котором учащийся вовлекается во взаимодействие со средой, моделирующей реальные процессы, управляет ее поведением, отвечает на сложные учебные запросы. Таким образом, обучающий тест может использоваться для построения электронного тьютора, модуля с высоким уровнем интерактивности и мультимедийности. Повышение интеллекта данного модуля обеспечивается новыми возможностями по более полному разбору ответа студента и увеличением возможного числа реакций системы на действия студента. Аттестующий режим тестирования должен использоваться как для проведения текущего контроля успеваемости в течение семестра, так и для проведения экзаменов и зачетов.

Кроме тестов, к контролирующим элементам относятся виртуальные лабораторные работы. Виртуальная лаборатория является модулем системы дистанционного обучения, представляющим собой обучающий комплекс, позволяющий студенту производить эксперименты либо с математической моделью, либо с физической установкой. Выполнение лабораторной работы заканчивается представлением отчета, который может быть проверен автоматически. В частном случае результатом выполнения лабораторной работы может быть формальное описание какой-либо системы, которая оценивается по реакциям на эталонные воздействия. Использование виртуальной лаборатории требуется в случае, когда невозможно реализовать авторский замысел средствами других видов электронных элементов системы – например, когда существует достаточно большое количество правильных ответов или задача проверки результата не является алгоритмической. Как и в случае с тестами, результат выполнения лабораторной работы доступен и студенту, и преподавателю сразу после ее окончания.

Еще одним контролирующим элементом является электронный практикум. Практикум содержит набор заданий, которые необходимо выполнить студенту. Предъявляемое задание выбирается из базы данных и закрепляется за конкретным студентом. В отличие от тестов, задание, которое предъявляется студенту в рамках практикума, не требует мгновенного выполнения. Системой определяется срок, в течение которого задание должно быть сдано. Результатом выполнения задания должен быть файл, отсылаемый студентом в базу данных. Проверка результата работы студента осуществляется преподавателем, который может поставить оценку или отправить работу на исправление, указав выявленные недостатки, не позволяющие ее принять. При неудовлетворительной оценке студенту может быть выдан другой вариант задания. Данный модуль

системы ДО может использоваться при организации таких видов учебной работы, как курсовой проект (работа), расчетно-графические работы, реферат.

Полный список, подробное описание и форматы представления элементов системы ДО AcademicNT приводится в «Техническом руководстве по разработке учебно-методического комплекса для системы дистанционного обучения», текущая версия которого публикуется на сайте системы дистанционного обучения СПбГУ ИТМО по адресу <<http://de.ifmo.ru/>>.

Участники учебного процесса, работая в системе ДО, кроме учебно-методических материалов, могут использовать различные сетевые средства электронного общения для оперативного обмена информацией, права доступа к которым также определяются ролью пользователя и параметрами, которые устанавливает владелец ресурса. В системе встроены доски объявлений, форумы, чаты и электронная почта. Следует отметить, что в рамках, например, форума студенты общаются не только с преподавателем, но и между собой, что еще более повышает эффективность такого инструмента.

Система ДО имеет широкие возможности по мониторингу учебного процесса. Возможность доступа к соответствующей информации и ее объем определяются ролью пользователя в системе. Одним из главных элементов мониторинга учебного процесса в системе ДО является электронный журнал. На страницах электронного журнала можно получить информацию как об успеваемости в какой-либо группе, так и о результатах определенного студента. При этом в журнале учитываются не только результаты электронного контроля с помощью тестов, практикумов и виртуальных лабораторий, но и текущий контроль успеваемости, предоставляемый вручную тьютором, ведущим занятия со студентами. Для оценки успеваемости студентов используется модульно-рейтинговый принцип. Работа студента с УМК оценивается состоянием в курсе и рейтингом. Рейтинг определяет, насколько качественно был усвоен изучаемый материал, а состояние позволяет оценить степень изучения курса. Состояния, по сути, являются показателем компетенций. С помощью электронного журнала тьютор не только отслеживает успеваемость, но и управляет учебным процессом.

Кроме информации об успеваемости, в системе ДО можно получить самую различную статистическую информацию. Уже разработано большое число различных форм статистической отчетности, позволяющих получить исчерпывающую информацию о зарегистрированных в системе подразделениях, пользователях и группах пользователей, о специальностях, по которым проводится обучение, об использовании УМК и их элементов в учебном процессе, о времени работы пользователей в системе, о результатах аттестаций по дисциплине, курсу или конкретному элементу. Отчеты дают информацию как о результатах отдельного пользователя, так и о положении в учебной группе, на кафедре, факультете и в Университете в целом. Формы используемых отчетов легко корректируются и добавляются в систему по мере необходимости. Такой подход к получению информации делает учебный процесс в системе ДО открытым и прозрачным.

Средства администрирования системы обеспечивают управление учебным процессом в системе ДО, содержанием УМК и собственно работой системы. Администрирование УМК позволяет непосредственно через Интернет осуществлять корректировку содержания элементов УМК. Важно отметить, что внесение изменений в УМК возможно непосредственно в ходе учебного процесса, что существенно повышает оперативность работы автора курса. Раздел управления учебным процессом определяет назначение групп студентов на учебные планы и программы дисциплин и задает время и место проведения электронного контроля успеваемости посредством формирования электронных ключей.

Заключение

При разработке системы ДО AcademicNT была решена задача обеспечения возможности использования новых педагогических инструментов, которые становятся доступными благодаря компьютерным технологиям – интерактива, мультимедиа, моделинга и коммуникативности. Практика использования системы в учебном процессе показала полное соответствие системы предъявляемым требованиям. Вместе с тем продолжилось совершенствование программно-аппаратного обеспечения системы. Для повышения надежности и защищенности сеансов работы пользователей система ДО переведена на использование протокола HTTPS, который обеспечивает проверку подлинности пользователей и позволяет установить безопасное подключение между отправителем и получателем. Временный разрыв связи, выключение клиентского компьютера теперь не приводит к потере сеанса при аттестации, и при повторном подключении работа пользователя будет продолжаться, а вся информация сохраниться. Кроме использования системы ДО AcademicNT в учебном процессе Университета, были разработаны схемы, обеспечивающие проведение самообследования и олимпиад, что подтвердило гибкость и универсальность разработанной системы. Все изменения и дополнения вносились в систему ДО оперативно, без остановки учебного процесса. И это очень большое преимущество системы ДО AcademicNT. Это показывает возможность оперативной адаптации системы к любым изменениям в организации учебного процесса Университета, что является залогом ее успешного использования в течение длительного времени.

Использование системы ДО AcademicNT выявило и ряд проблем, требующих обязательного решения. Основная проблема заключается в том, что громадные возможности этой информационной системы не используются в полной мере. Количество УМК, разработанных с 2003 г., недостаточно и покрывает лишь небольшую часть учебного процесса Университета. А ведь именно использование системы ДО позволит упростить процесс перехода на модульные принципы. При разработке УМК не в полной мере используются возможности системы и компьютерных технологий. Слабо используются обучающие возможности системы, мало реализовано схем интерактивных диалогов, практически не используются возможности программирования состояний в электронном курсе. Часто отсутствует оперативный контроль за результатами аттестаций как со стороны преподавателей, ведущих дисциплину, так и со стороны сотрудников деканатов. Преподавателями не в полной мере используются возможности, которые им дает электронный журнал, средства электронного общения. В этой связи крайне важным представляется подготовка авторов и тьюторов, готовых в полной мере использовать современные компьютерные технологии, которые предоставлены Университетом.

Литература

1. Васильев В.Н., Лисицына Л.С., Лямин А.В., Шехонин А.А. Система дистанционного обучения СПбГИТМО: Современное состояние и перспективы развития // Современные образовательные технологии: Сборник статей / Под ред. В.Н. Васильева, Ю.Л. Колесникова – СПб: СПбГИТМО(ТУ), 2001. – С. 13–34.
2. Васильев В.Н., Шехонин А.А., Лямин А.В., Тарлыков В.А. Развитие системы дистанционного обучения в соответствии с целевой комплексной программой университета // Научно-технический вестник Санкт-Петербургского государственного института точной механики и оптики (технического университета). 2001. № 1. С.45–49.
3. Лямин А.В., Тарлыков В.А., Шехонин А.А. Методология построения системы дистанционного обучения в СПбГИТМО(ТУ) на основе интернет-технологий // Сбор-

- ник трудов конференции «Оптика и образование». – Санкт-Петербург, 2002. – С. 20–22.
4. Васильев В.Н., Шехонин А.А., Лямин А.В., Тарлыков В.А. Информационно-технологическое сопровождение учебного процесса в СПбГУ ИТМО // Сборник трудов конференции «Оптика и образование – 2004». – СПб: СПбГУ ИТМО, 2004. – С. 62–63.
 5. Шехонин А.А., Лямин А.В., Тарлыков В.А., Чежин М.С. Организационно-методическое сопровождение текущего контроля успеваемости в СПбГУ ИТМО на базе компьютерного тестирования // Материалы XI международной конференции. Современные технологии обучения: международный опыт и российские традиции «СТО-2005». – Санкт-Петербург, 2005. – Т. 1. – С. 14–15.
 6. Лямин А.В., Чежин М.С. Модульное построение электронных учебно-методических комплексов для системы дистанционного обучения // Труды XII Всероссийской научно-методической конференции «Телематика'2005». – Санкт-Петербург, 2005. – Т. 2. – С. 511–512.

ПРИМЕРЫ РЕАЛИЗАЦИИ СЕТЕВЫХ ВИРТУАЛЬНЫХ ЛАБОРАТОРИЙ В СРЕДЕ СИСТЕМЫ ДИСТАНЦИОННОГО ОБУЧЕНИЯ

О.Е. Вашенков, А.А. Волкова, А.В. Лямин

В работе рассмотрены примеры реализации интерактивных аттестующих элементов с автоматической проверкой ответов – виртуальных лабораторных работ. Доступ к таким элементам и их функционирование обеспечивается средствами системы дистанционного обучения. Рассмотрены два примера реализации виртуальных лабораторных работ по дисциплинам «Когерентная оптика» и «Теоретическая механика», особенности взаимодействия лабораторных установок с системой дистанционного обучения, принципы формирования эталонных данных для проверки и интерфейс пользователя.

Введение

В системе дистанционного обучения AcademicNT, разработанной в (СПбГУ ИТМО), определено два типа элементов, которые могут использоваться как для обучения, так и для аттестации с проверкой ответов в автоматическом режиме: электронные тесты и виртуальные лабораторные работы [1–3]. Электронные тесты состоят из информационных кадров и заданий. Множество правильных ответов каждого тестового задания является разрешимым, т.е. задача определения принадлежности ответа данному множеству является алгоритмически разрешимой. К подобным заданиям относятся все известные формы тестовых заданий – открытой формы, закрытой формы, на установление правильной последовательности и на установление соответствия [4]. Из перечисленных форм самой сложной является открытая форма. Эта форма требует тщательного согласования формулировки задания с возможностями, разрабатываемого автором, анализатора ответа. В большинстве случаев анализатор ответов строится на основе регулярных выражений [5], ограничения которых хорошо известны [6]. Поэтому задача проверки заданий даже со счетным множеством, имеющим достаточно большое количество элементов, становится трудоемкой. Примерами подобных заданий могут служить:

- разработка электронной схемы, осуществляющей выделение амплитуды гармонического сигнала;
- построение механической системы, осуществляющей преобразование поступательного движения во вращательное;
- написание компьютерной программы, которая реализует заданный алгоритм.

В случае, когда множество правильных ответов на задание является бесконечным или неразрешимым, можно использовать элемент системы, который называется «Виртуальная лаборатория». В виртуальных лабораторных работах правильный ответ задается множеством прецедентов «вход–выход». Ответ на каждое задание рассматривается как система, которая имеет вход и выход. На вход системы подается эталонный входной сигнал, снимается выходной сигнал, который сверяется с эталонным выходным сигналом. Если выходной сигнал совпадает с эталонным выходным сигналом с заданной точностью, то ответ признается верным.

Схема проверки результатов выполнения виртуальных лабораторных работ

Программный инструмент виртуальных лабораторий хранится в базе данных системы дистанционного обучения. Пользователи системы получают доступ к виртуальным лабораторным работам в соответствии с регламентом, установленным в системе. Доступом к системе управляет Web-сервер на основе информации, хранящейся в базе данных. Проверка результатов лабораторных работ осуществляется автономными

серверами, на которых установлено специальное программное обеспечение. Модули, обеспечивающие функционирование виртуальных лабораторий, показаны на рис. 1.

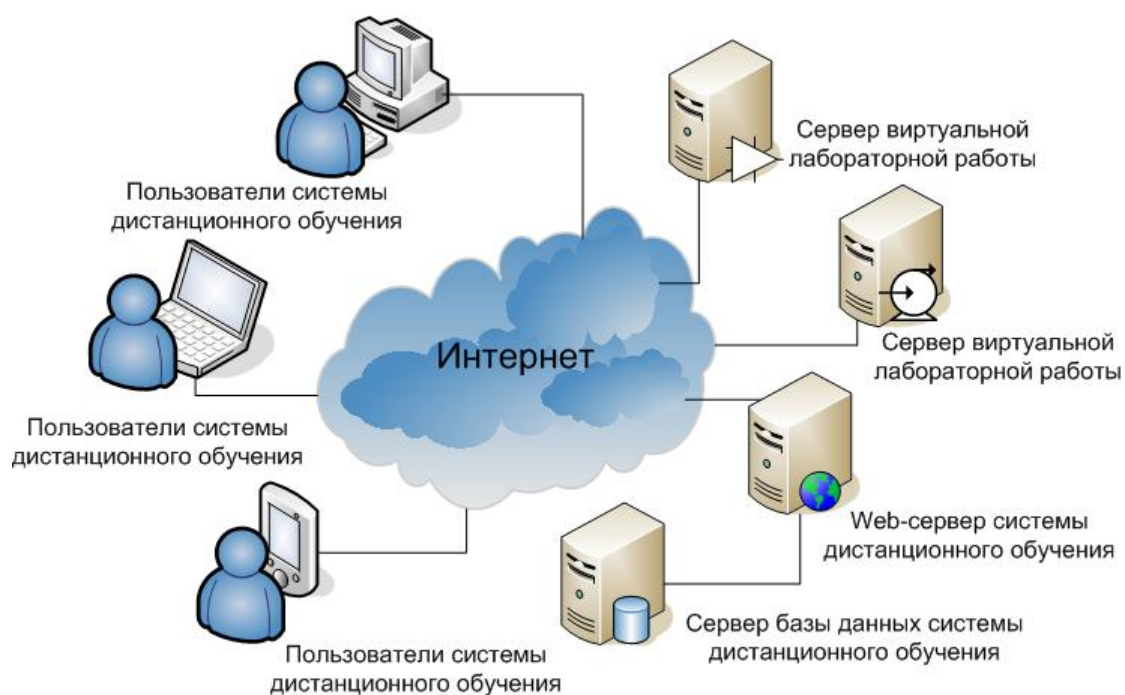


Рис. 1 Общая схема доступа к виртуальным лабораториям

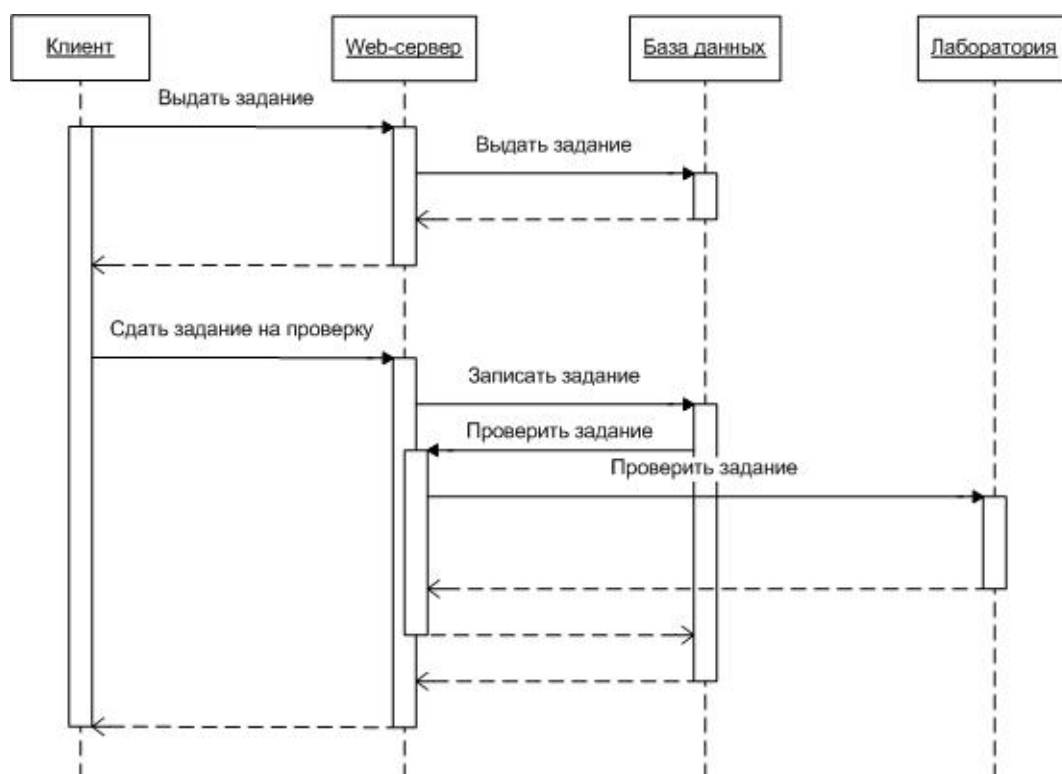


Рис. 2. Схема проверки ответа

На рис. 2 показана последовательность, которая выполняется во время проверки ответа. Ответ клиента – пользователя системы поступает на Web-сервер системы, который фиксирует его в базе данных, получает из базы данных прецеденты «вход–выход»

и вместе с ответом передает их на проверяющий сервер данной лаборатории. После получения ответа от проверяющего сервера инициализируется алгоритм вычисления оценки.

Виртуальная лаборатория по дисциплине «Когерентная и нелинейная оптика»

Виртуальная лабораторная работа по дисциплине «Когерентная и нелинейная оптика» была разработана профессором кафедры лазерной техники и биомедицинской оптики СПбГУ ИТМО В.А. Тарлыковым. Для дисциплины «Когерентная и нелинейная оптика» центром дистанционного обучения СПбГУ ИТМО было реализовано три виртуальные лабораторные работы на языке программирования Java [7]:

- «Исследование степени когерентности лазерного излучения»;
- «Исследование степени когерентности некогерентного источника»;
- «Исследование влияния размеров апертуры на контраст спекл-картины».

Для трех виртуальных лабораторных работ был реализован один проверяющий сервер, выполняющий расчеты в зависимости от типа работы. Исходными данными для проверяющего сервера служат:

- ответ студента;
- допустимая погрешность;
- интервал значений, для которого требовалось построить график;
- минимально-допустимое количество точек.

Интерфейс пользователя для работы «Исследование степени когерентности лазерного излучения» приведен на рис. 3 и 4. Интерфейс построения графиков, представленный на рис. 3, использовался во всех трех работах.

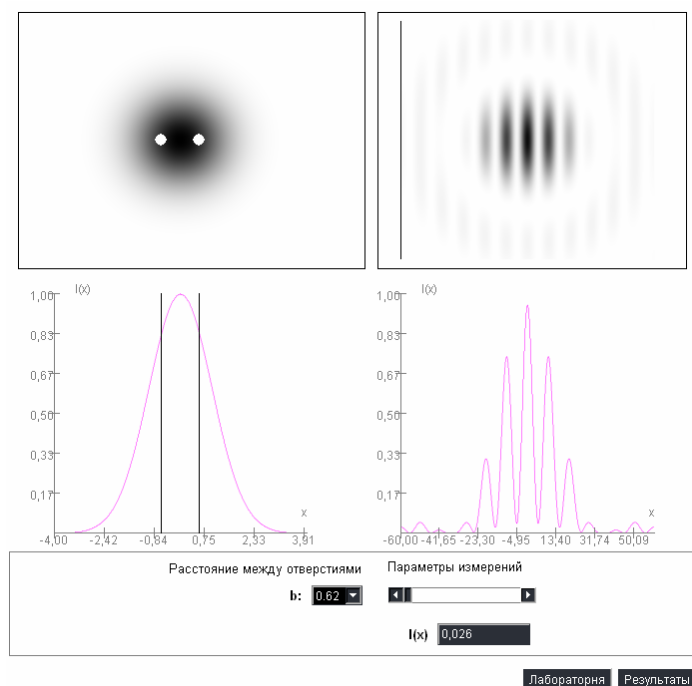


Рис. 3. Лабораторная работа «Исследование когерентности лазерного излучения»

Для построения графической зависимости в этой работе, студент измеряет значение интенсивности $I(x)$ на дифракционной картине для заданного расстояния между отверстиями b . Изменение координаты x в плоскости дифракционной картины происходит при перемещении слайдера.

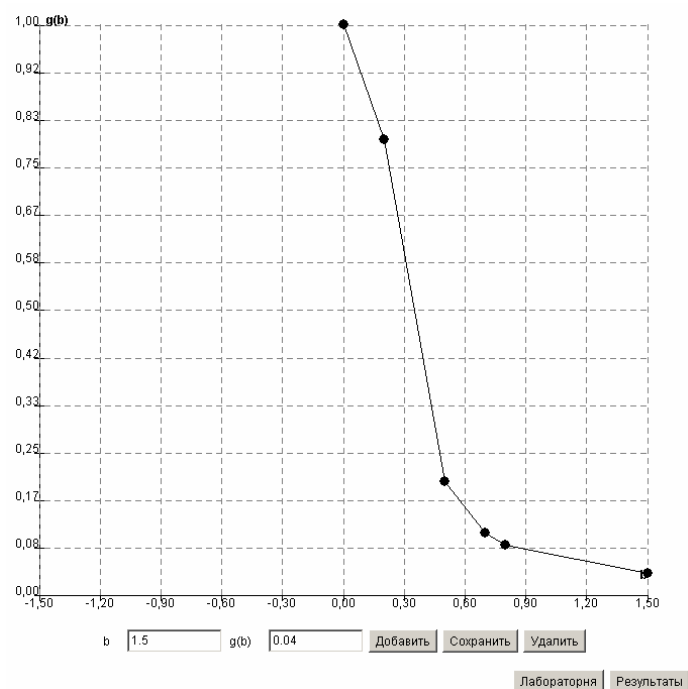


Рис. 4. Оформление ответа в графической форме

Интерфейс пользователя лабораторной работы «Исследование степени когерентности некогерентного источника» представлен на рис. 5.

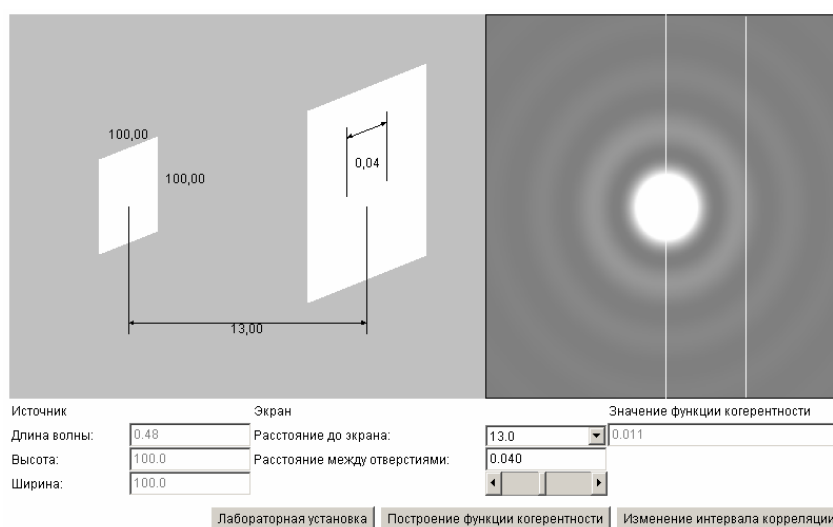


Рис. 5. Лабораторная работа "Исследование когерентности некогерентного источника"

В данной работе пользователю требуется измерить значение функции когерентности на различном расстоянии от центра дифракционной картины. Ответом служит ряд измерений для трех различных расстояний до плоскости наблюдения.

Интерфейс пользователя лабораторной работы «Исследование влияния размеров апертуры на контраст спекл-картины» представлен на рис. 6 и 7.

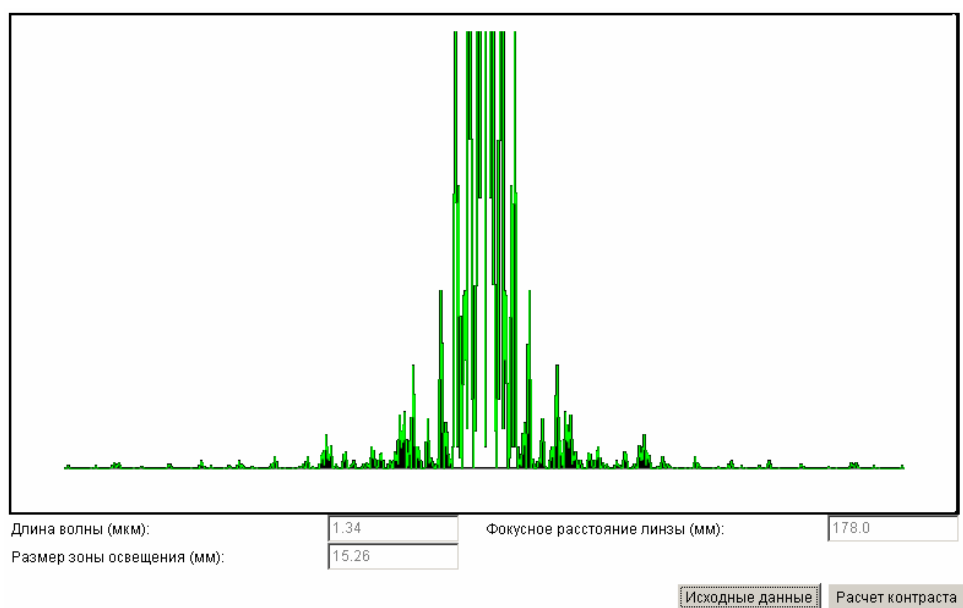


Рис. 6. Представление исходных данных для лабораторной работы

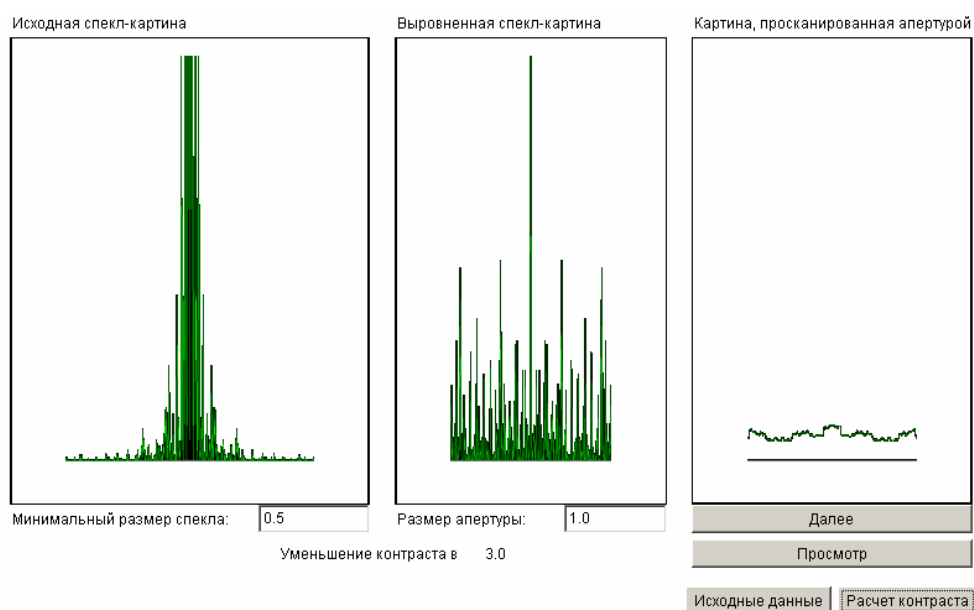


Рис. 7. Лабораторная работа «Исследование влияния размеров апертуры на контраст спекл-картины»

В данной работе студенту требуется определить размер апертуры для уменьшения контраста спекл-картины в заданное число раз. Расчет требуется провести для нескольких значений контраста. По завершению расчетов выводится график зависимости контраста от указанных значений размеров апертуры. Затем ответ передается на проверяющий сервер.

Виртуальная лаборатория по дисциплине «Теоретическая механика»

Виртуальная лабораторная работа по дисциплине «Теоретическая механика» разработана заведующим кафедрой теоретической и прикладной механики СПбГУ ИТМО В.Г. Мельниковым. Визуальная часть виртуального лабораторного комплекса реализована центром дистанционного обучения СПбГУ ИТМО в виде апплета, выполненного

при помощи языка программирования Java 2 [8]. Виртуальная установка представляет собой сложную двухстепенную механическую систему, разработанную на кафедре теоретической механики, анимация движения которой осуществляется по законам механики. Математическая модель нелинейных колебаний системы выражается через систему нелинейных дифференциальных уравнений с заданными начальными условиями, которая решалась при помощи метода Рунге-Кутты четвертого порядка точности.

Общая схема установки приведена на рис. 8. Это механическая система с двумя степенями свободы. Тело массой m_{12} катится вдоль рельса при помощи трех колес подвеса радиусом r_1 и массой m_{11} , совершая вынужденные колебания. Положение тела в каждый момент времени относительно точки отсчета O определяет координата x , линейная скорость тела обозначается V . Коэффициент трения между колесом подвеса и рельсом равен K_4 . Пружина с коэффициентом жесткости K_1 разгоняет тело к положению равновесия. Справа к установке присоединен демпфер, создающий силу сопротивления, пропорциональную скорости поршня. Коэффициент трения демпфера равен K_3 . К левой вертикальной стенке прикреплен электромотор. На валу мотора под прямым углом к оси вращения закреплен однородный стержень длиной r_3 , к концу которого прикреплена пружина с коэффициентом жесткости K_2 . Другой конец пружины закреплен на теле массой m_{12} . Вал электромотора равномерно вращается с угловой скоростью $\beta = pt$, где p – частота вращения ротора двигателя.

К телу прикреплен маятник массой m_{21} с грузом массой m_{22} . Маятник длиной l вращается в подшипнике с коэффициентом вязкого трения K_5 . Груз, имеющий радиус инерции i_{22} , неподвижно прикреплен к маятнику на расстоянии r от оси вращения.

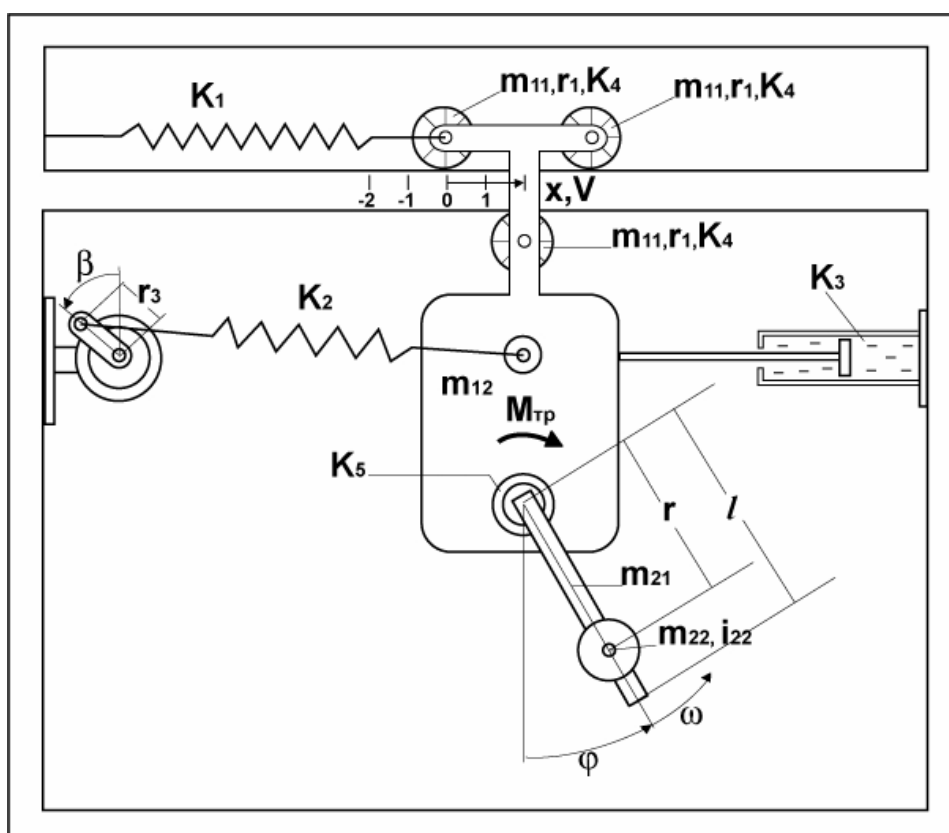


Рис. 8. Схема лабораторной установки дисциплине «Теоретическая механика»

Интерфейс апплета виртуального лабораторного стенда как ключевой элемент человеко-машинного взаимодействия можно мысленно разделить на три основных части:

- область управления;
- область определения начальных условий;
- графическое представление установки.

На области управления расположены кнопки запуска и остановки лабораторной установки, текстовое поле для изменения шага дискретизации, кнопка вызова окна настроек, кнопка возвращения результата эксперимента, а также элементы управления показом окон выводимых графиков. После щелчка на кнопке «Пуск» установка начинает движение, и все элементы для изменения параметров оказываются заблокированы. Разблокировка происходит только после щелчка на кнопке «Стоп». Программой предусмотрено вывод пяти графиков.

Окно настроек предназначено для управления анимацией и временем проведения эксперимента, изменения масштаба осей выводимых графиков, а также для выбора используемых степеней свободы при проведении эксперимента – вращательной и поступательной, только вращательной, только поступательной и нестационарной.

На области определения начальных условий расположены элементы для изменения начальных условий и изменяемых параметров.

Интерфейс виртуального лабораторного стенда предусматривает следующие шаги сценария поведения пользователя при работе с ресурсом:

- корректировка начальных параметров модели;
- выбор просматриваемых графиков;
- запуск установки;
- остановка установки;
- анализ полученных результатов;
- при необходимости корректировка начальных параметров и повторный запуск установки;

При повторном запуске время начинает отсчитываться с нуля, а предыдущие графики стираются.

Заключение

В данной статье была продемонстрирована возможность использования широко распространенного в программировании способа контроля качества, основанного на тестировании. В случае, когда множества правильных ответов является бесконечным или неразрешимым, данный способ проверки ответов представляется единственным физически реализуемым. Описанные в статье примеры использования подхода иллюстрируют его эффективность при построении виртуальных лабораторий для двух дисциплин – «Когерентная и нелинейная оптика» и «Теоретическая механика».

Литература

1. Лямин А.В., Чежин М.С. Организация учебного процесса университета на базе использования современных компьютерных технологий // Учебно-методический сборник. Выпуск 1. - СПб: СПбГУ ИТМО, 2005. С. 63–77.
2. Лямин А.В., Чежин М.С. Модульное построение электронных учебно-методических комплексов для системы дистанционного обучения // Труды XII Всероссийской научно-методической конференции «Телематика'2005». – Санкт-Петербург, 2005. - Т. 2. - С. 511–512.

3. Лямин А.В., Чежин М.С. Система дистанционного обучения СПбГУ ИТМО // Труды II Международной конференции «Информационные системы для HRM и обучения». – Санкт-Петербург, 2005. – С. 80-83.
4. Аванесов В.С. Композиция тестовых заданий. – М.: Центр тестирования, 2002. – 240 с.
5. Агапонов С.В. и др. Средства дистанционного обучения. Методика, технология, инструментарий. Серия «Мастер решений». – СПб.: БХВ-Петербург, 2003. – 336 с.
6. Альфред Ахо, Рави Сети, Джеффри Ульман. Компиляторы. Принципы, технологии, инструменты. – М.: Вильямс, 2003. – 768 с.
7. Вашенков О.Е., Лямин А.В., Тарлыков В.А. Оценивание результатов обучения в среде электронного учебно-методического комплекса по дисциплине «Когерентная оптика» // Сб. трудов конф. «Оптика и образование - 2006». Под общ. редакцией проф. А.А. Шехонина. – СПб: СПбГУ ИТМО, 2006. – С.70–71.
8. Волкова А.А. Технология построения виртуальных лабораторных работ / Сборник материалов Всероссийского конкурса инновационных проектов аспирантов и студентов по приоритетному направлению развития науки и техники «Информационно-телекоммуникационные системы» / Под. ред. А.О. Сергеева. – М.: ГНИИ ИТТ «Информика», 2005. – С. 98.

РАЗРАБОТКА СРЕДЫ ДЛЯ ВИЗУАЛЬНОГО ПРОГРАММИРОВАНИЯ СТРУКТУРЫ ЭЛЕКТРОННЫХ КУРСОВ

А.В. Лямин, М.В. Плешкова

Статья посвящена описанию структуры электронного курса в системе ДО СПбГУ ИТМО AcademicNT, основанного на компетентностном модульно-рейтинговом подходе. Рассмотрены состояния компетенций курса и определены формулы для расчета состояний электронного курса. Описано программирование структуры электронного курса на основе конечно-автоматного подхода. Приведены основные возможности среды для визуального программирования структуры электронных курсов системы ДО.

Введение

Система дистанционного обучения СПбГУ ИТМО AcademicNT основана на сетевой Интернет-технологии, в которой могут быть реализованы различные способы и методы обучения – электронные учебники с удаленным доступом, виртуальные лаборатории, электронные практикумы, электронные библиотеки, телеконференции студентов и преподавателей, тестирующие обучающие и аттестующие системы. Электронный учебно-методический комплекс по дисциплине в системе построен на основе компетентностного модульно-рейтингового подхода и имеет иерархическую структуру, изображенную на рис. 1 [1, 2]. Описание структуры электронного курса содержит оглавление курса, список элементов курса; правила формирования рейтинга, правила фиксации состояний компетенций, правила организации работы с курсом.

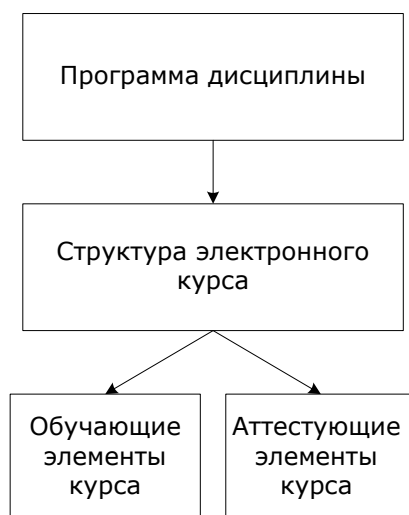


Рис. 1. Структура учебно-методического комплекса

Описания состояний компетенций курса

Любой курс можно представить в виде последовательности групп, каждая из которых содержит набор аттестующих и обучающих элементов. Внутри группы задается либо последовательное прохождение элементов, либо параллельное. На рис. 2 показаны две группы элементов – G_1 и G_2 , в группе G_1 предусмотрено параллельное прохождение элементов X_1 и X_2 , в группе G_2 – последовательное прохождение элементов Y_1 и Y_2 .

Состояния компетенций определяются как требования к знаниям и практическим умениям, которые должны быть сформированы у слушателя при освоении соответствующих компетенций [3]. Правила фиксации состояний компетенций показывают, какой уровень освоения материала достиг слушатель. На рейтинг слушателя могут влиять следующие параметры: оценка, полученная при аттестации, номер попытки, время сдачи аттеста-

ции. Данные параметры программируются с помощью переменных курса. Рейтинг можно изменить при записи значения выше, чем текущее значение переменной. При попытке записать в переменную значение, которое меньше текущего, рейтинг не изменится. Для слушателя, который исчерпал попытки, выделенные на аттестацию, может быть предусмотрена дополнительная попытка за счет введения специального состояния.

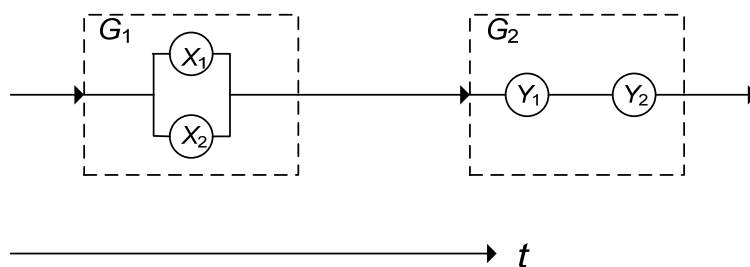


Рис. 2. Последовательность групп в курсе

Количество состояний при параллельном прохождении курса вычисляется по формуле $S = 2^n$, где n – количество аттестующих элементов. При последовательном прохождении количество состояний равно $S = n + 1$. Данные формулы не учитывают состояния для дополнительных попыток на аттестацию, для блокирования входа в элементы курса во время аттестации.

Программирование структуры электронного курса

Программирование структуры электронного курса осуществляется на основе конечно-автоматного подхода. Автомат переходов имеет одно входное состояние, которое определяет начало обучения, и одно выходное состояние, которое свидетельствует о получении результирующей компетенции. На рис. 3 приведен пример диаграммы переходов для курса, в который входят три аттестующих элемента.

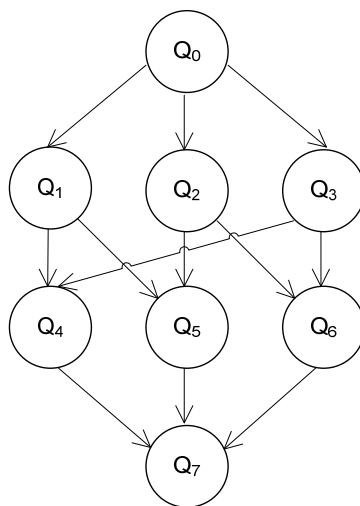


Рис. 3. Пример диаграммы переходов

Состояние Q_0 – это состояние начала обучения, в котором доступны для сдачи аттестации. При успешной сдаче одной аттестации система переходит в состояния Q_1 , Q_2 , Q_3 . В этих состояниях доступны для сдачи две следующих аттестации, прохождение которых переводит слушателя в одно из следующих состояний Q_4 , Q_5 или Q_6 . Состояние Q_7 – это состояние завершения обучения, в которое система переводит слушателя после сдачи всех аттестаций.

Множество переменных курса можно описать синтаксическим деревом. Пример синтаксического дерева приведен на рис. 4. Переменная «Оценка за курс» является корневой для всех остальных переменных, и ее значение вычисляется с помощью функции, которая применяется к дочерним переменным – листьям синтаксического дерева. В AcademicNT определены пять функций: max – максимум, min – минимум, sum – сумма, avg – среднее, mul – произведение.

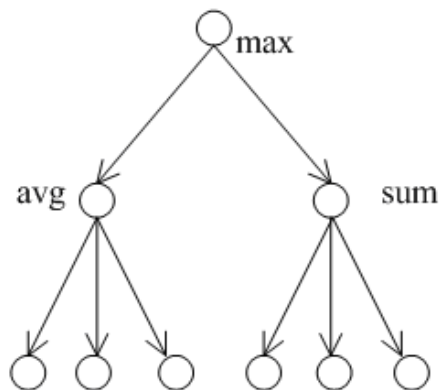


Рис. 4. Пример построения синтаксического дерева переменных курса

Можно показать, что для представления произвольного кусочно-гладкого отображения достаточно использовать только три функции – max, sum и mul. Действительно любая гладкая функция представима рядом Тейлора $F_n(x) = a_0 + a_1x + a_2x^2 + \dots + a_{n-1}x^{n-1} + a_nx^n$. Таким образом, описанный механизм осуществляет кусочно-гладкое отображение результатов работы слушателя с курсом в рейтинг.

Заключение

В соответствии с вышеизложенными правилами была разработана среда для визуального программирования структуры электронных курсов. С помощью данной среды автор может создавать группы и определять связи между ними, добавлять элементы и указывать их атрибуты, такие как название элемента, количество попыток прохождения аттестаций. Для каждого аттестующего элемента устанавливаются правила оценивания и условия перехода к следующим элементам. Также существует возможность создавать и изменять дерево переменных, причем при добавлении новой переменной связи с элементами пересчитываются автоматически.

Литература

1. Лямин А.В., Чежин М.С. Модульное построение электронных учебно-методических комплексов для системы дистанционного обучения // Труды XII Всероссийской научно-методической конференции «Телематика'2005». – Санкт-Петербург, 2005. – Т. 2. – С. 511–512.
2. Лямин А.В., Чежин М.С. Система дистанционного обучения СПбГУ ИТМО // Труды II Международной конференции «Информационные системы для HRM и обучения». – Санкт-Петербург, 2005. – С. 80–83.
3. Васильев В.Н., Лисицына Л.С., Лямин А.В. Сетевое сообщество на основе методического Интернет-центра: первый опыт создания и перспективы развития / Научно-технический вестник СПбГУ ИТМО. 2006. Вып. 32 «Информационные технологии: теория, методы, приложения». С. 254–259.

ПОДХОД К СОЗДАНИЮ СРЕДСТВ ТЕМАТИЧЕСКОЙ КАТЕГОРИЗАЦИИ ИНТЕРНЕТ-РЕСУРСОВ

С.В. Свечников

Предложен подход для создания алгоритмов и средств поиска, анализа и тематической категоризации интернет-ресурсов. Представлено решение задач индексации и автоматической категоризации ресурсов сети Интернет за счет выделения терминов и присвоения им весовых коэффициентов, что позволяет достаточно быстро и эффективно оценить контент интернет-ресурса. Рассмотрены вопросы, связанные с оценкой качества категоризации, задачами сбора и обработки контента интернет-ресурсов.

В настоящее время в России существует необходимость в применении систем тематической категоризации, а вместе с ними и систем для управления доступом к интернет-ресурсам. Открытое информационное пространство содержит большое количество сайтов различного содержания, и наряду с полезной информацией Интернет содержит ресурсы, объективно опасные для нравственного здоровья общества, оказывающие негативное воздействие, в первую очередь, на подрастающее поколение. Российский сегмент сети – один из самых быстроразвивающихся, количество пользователей Интернета в России по различным данным достигает 25 миллионов человек, из них 2 миллиона детей [1]. Обеспечение учебных заведений и публичных библиотек доступом к сети Интернет увеличивает количество учащихся, пользующихся различными сервисами и информационными источниками, предоставляемыми глобальной сетью. Такой бесконтрольный доступ к сети Интернет может привести к серьезным угрозам для детей и учащихся. Также Интернет бесконтрольно используется в личных целях работниками умственного труда, имеющими доступ к глобальной сети, что снижает эффективность их работы и производительность корпоративной сети [2]. При этом методы прямого регулирования (цензуры) неэффективны, встречаются протест пользователей и юридически несостоятельны, поскольку противоречат естественным правам граждан на свободу высказываний и волеизъявления.

В связи с этим решение этой проблемы надо искать не в цензуре, а в предоставлении инструмента для защиты от нежелательного контента, который пользователи могут использовать по своей воле и по своему усмотрению [3]. Таким инструментом является разрабатываемая система тематической категоризации интернет-ресурсов.

Реализация системы тематической категоризации интернет-ресурсов предполагает решение следующих задач:

- индексация интернет-ресурсов (преобразование интернет-ресурсов к единому формату);
- автоматическая категоризация интернет-ресурсов, обучение системы и отнесение текстовой информации к заранее определенной категории;
- оценка качества категоризации с использованием метрик из информационного поиска.

Представленные подзадачи связаны, в первую очередь, с анализом текстовой информации веб-страницы, т.е. ее содержанием (контентом).

Пусть дано множество интернет-ресурсов D , разделенное на два непересекающихся подмножества T_r и T_s , называемых обучающей и тестовой выборкой. На основании обучающей выборки строится классификатор категорий, а на тестовой выборке проверяется качество категоризации. Пусть также дано соответствие между интернет-ресурсами и некоторой категорией c в виде $\Phi : D \rightarrow \{0,1\}$, устанавливающее для каждого интернет-ресурса значение 1, в случае принадлежности интернет-ресурса категории, и 0 – в противном случае [3–6].

Необходимо построить, используя только информацию из обучающей выборки T_r , функцию $\Phi' : D \rightarrow \{0,1\}$, аппроксимирующую Φ , чтобы число ошибок E на тестовой выборке T_s было наименьшим.

$$E = \sum_{T_s} |\Phi - \Phi'| \rightarrow \min. \quad (1)$$

Пусть T – множество терминов, каким-либо образом выделенное из интернет-ресурсов категории c . Тогда интернет-ресурс можно представить в виде терминологического вектора

$$d_j = (w_{1j}, \dots, w_{|T|j})^T, \quad (2)$$

где w_{ij} – вес термина t_i в интернет-ресурсе d_j .

Описания каждой из категорий представим в виде векторов той же размерности, что и вектора интернет-ресурсов:

$$c = (c_1, \dots, c_{|T|})^T, \quad (3)$$

где c_i – вес термина t_i в описании категории c .

При таком подходе существуют два ограничения:

- нет дополнительной информации о категориях, к которым прикрепляются интернет-ресурсы;
- нет никакой внешней информации о интернет-ресурсе, кроме той информации, которая содержится в нем.

Каждый интернет-ресурс – это вектор, где номера терминов (слов) – его координаты, а веса терминов – значения координат, размерность вектора – это количество терминов, встречающихся в интернет-ресурсе. Так как учитываются все термины, вектора получаются большого размера, что затрудняет процесс индексации, поэтому необходимо уменьшить размерность вектора. Для уменьшения размерности вектора в качестве терминов используем не слова, а устойчивые словосочетания, не учитываем редкие слова, которые не несут полезной информации, не рассматриваем часто встречающиеся слова.

Процесс индексации представим следующим образом (рис. 1):

- очистка страницы;
- выделение терминов;
- исключение терминов, не несущих смысловой нагрузки;
- замена общих и специфичных терминов;
- присваивание терминам весовых коэффициентов.

Сначала проводится очистка страницы интернет-ресурса, т.е. удаляется навигационная часть, теги html, скрипты, стоп-слова – частотные слова языка, не несущие смысловой нагрузки (предлоги, союзы, частицы, местоимения, некоторые глаголы), за счет этого уменьшается объем поисковой базы и повышается производительность поиска [7–9]. После этого в тексте с помощью функции анализа контента интернет-ресурса выделяются термины – логические выражения, состоящие из слов и словосочетаний, связанные операторами AND, OR, NOT. Для исключения терминов, не несущих смысловой нагрузки, используется пометка «исключение», которая показывает, что термин не относится к теме. Оставшиеся термины могут также обладать недостатками: существуют термины, которые слишком специфичны, или, наоборот, значение которых слишком общее, поэтому их необходимо заменить более подходящими. Это увеличивает полноту индексирования. Для замены специфичных терминов используется тезаурус RCO, который представляет собой словарь общей лексики с семантическими отношениями между словами [10]. Использование тезауруса повышает качество анализа текста и полноту поиска информации, позволяя расширять запрос синонимичными, более общими и более частными понятиями. Общие термины заменяются сочетаниями терминов или не-

сколькими связанными терминами, имеющими более определенное значение. После определения терминов необходимо провести лемматизацию – приведение терминов к нормальной форме (мужской род, единственное число). Тем самым уменьшается словарь терминов и повышается скорость работы индексации.

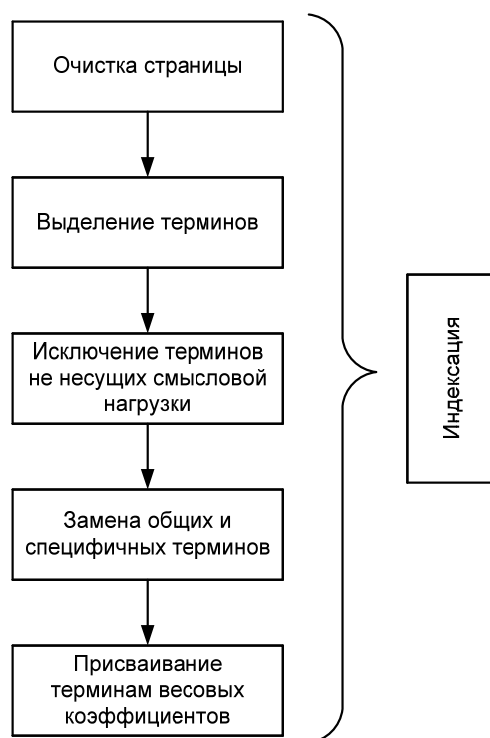


Рис. 1. Индексация интернет-ресурсов.

Заключительным этапом является присвоение терминам весовых коэффициентов. Исходное представление интернет-ресурса выглядит следующим образом: Интернет-ресурс = коллекция слов (терминов) T . Каждый термин $t_i \in T$ имеет определенный вес w_{ij} по отношению к интернет-ресурсу $d_j \in D$ (рис. 2), т.е. встречаемость этого слова на странице интернет-ресурса. Порядок слов учитывать не будем. На основании этих признаков каждому слову сопоставляется его вес.

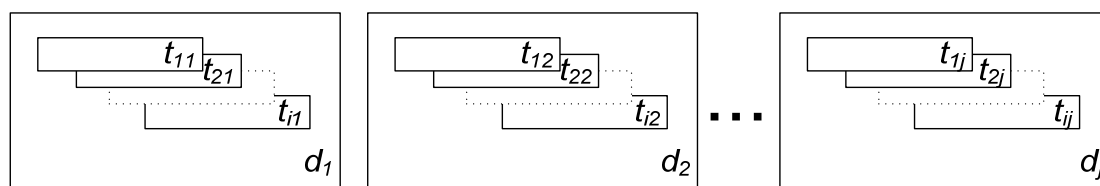


Рис. 2. Интернет-ресурсы, содержащие термины.

Таким образом, каждый ресурс можно представить в виде вектора весов его терминов $d_j = \{w_{1j}, \dots, w_{|T|j}\}$. Веса документов нормируем так, чтобы $w_{ij} \geq 0$ и $w_{ij} \leq 1$, где $i \in (0, |T|)$ и $j \in (0, |D|)$.

Для вычисления веса термина на странице интернет-ресурса используем классический частотный метод вычисления степени соответствия интернет-ресурса, так как этот метод относительно прост и имеет несложный алгоритм, что принципиально при обработке больших объемов документов.

Вычисляем вес термина следующим образом:

$$w_{ij} = tf_{ij} \cdot \frac{1}{df_j}, \quad (4)$$

где tf_{ij} (частота термина) – это отношение числа терминов t_i в интернет-ресурсе d_j к общему количеству терминов в этом интернет-ресурсе, таким образом, оценивается важность термина t_i в пределах одного интернет-ресурса:

$$tf_{ij} = \frac{T_{ij}}{|T_i|}, \quad (5)$$

где $j = 1, \dots, T$, $i = 1, \dots, D$, T_{ij} – число терминов t_i в интернет-ресурсе d_j , T_i – общее число терминов в интернет-ресурсе d_j . df_j (частота интернет-ресурса) – это отношение количества интернет-ресурсов категории, в которых встретился термин t_i , к общему количеству интернет-ресурсов категории:

$$df_j = \frac{|D_j|}{|D|}, \quad (6)$$

где $j = 1, \dots, T$, D_j – число интернет-ресурсов, в которых встретился термин t_i , D – общее количество интернет-ресурсов категории.

Таким образом, чем чаще термин встречается на странице интернет-ресурса, но реже встречается во всех интернет-ресурсах, тем выше будет его вес в данном интернет-ресурсе.

Наиболее трудоемкой частью реализации системы является разработка процесса, отвечающего за автоматическую категоризацию интернет-ресурсов, обучение системы на уже категоризированных интернет-ресурсах и определение соответствия категории. Алгоритм автоматической категоризации интернет-ресурсов заключается в следующем:

- вычисляется мера близости страницы интернет-ресурса и категории – степень соответствия ресурса категории;
- для каждой страницы выбирается категория, наиболее близкая к ресурсу;
- если значение степени соответствия ресурса превышает некоторое пороговое значение категории, ресурс добавляется в категорию;
- если значение степени соответствия ресурса не превысило порогового значения категории, ресурс не добавляется в категорию, и решение о принадлежности его к категории определяет эксперт.

Степень соответствия (CSV) между категорией c и интернет-ресурсом d_j определяем как скалярное произведение между их векторными представлениями:

$$CSV(c, d_j) = c \cdot d_j = \sum_i c_i d_{ij}, \quad (7)$$

Будем принимать решение о принадлежности интернет-ресурса к категории, если степень соответствия достигнет заданного порога τ . Таким образом, получаем:

$$\Phi'(c, d_j) = \begin{cases} 1, & CSV(c, d_j) \geq \tau \\ 0, & CSV(c, d_j) < \tau \end{cases}. \quad (8)$$

После того как вычислена степень соответствия между категорией и интернет-ресурсом, а также пороговое значение категории, необходимо провести обучение. Цель обучения – настройка весовых коэффициентов и порогового значения таким образом, чтобы процедура категоризации относилась положительные примеры к категории, а отрицательные примеры – не относилась, т.е. чтобы суммы весовых коэффициентов всех положительных примеров были равны либо превышали пороговое значение, а суммы для отрицательных примеров были ниже порога.

Для оценки качества категоризации интернет-ресурсов применяем метрики из информационного поиска, такие как полнота, точность, F-мера [11].

Пусть D_r – множество интернет-ресурсов, категоризированных экспертами, а D_a – множество интернет-ресурсов, категоризированных автоматически.

Полнота категоризации интернет-ресурсов по категории вычисляется как отношение количества правильно категоризированных интернет-ресурсов системой к общему числу интернет-ресурсов, относящихся к этой категории.

$$r = \frac{|D_a \cap D_r|}{|D_r|} \quad (9)$$

Точность категоризации интернет-ресурсов по категории вычисляется как отношение количества правильно категоризированных интернет-ресурсов системой к общему числу интернет-ресурсов, автоматически категоризированных системой.

$$p = \frac{|D_a \cap D_r|}{|D_a|} \quad (10)$$

Для идеального алгоритма полнота и точность должны быть равны 100%.

F-мера, т.е. сводная оценка качества категоризации, определяется как гармонически среднее полноты и точности:

$$F = \frac{2 \cdot r \cdot p}{r + p} \quad (11)$$

Алгоритм осуществления сбора и обработки данных интернет-ресурсов выглядит следующим образом (рис. 3).

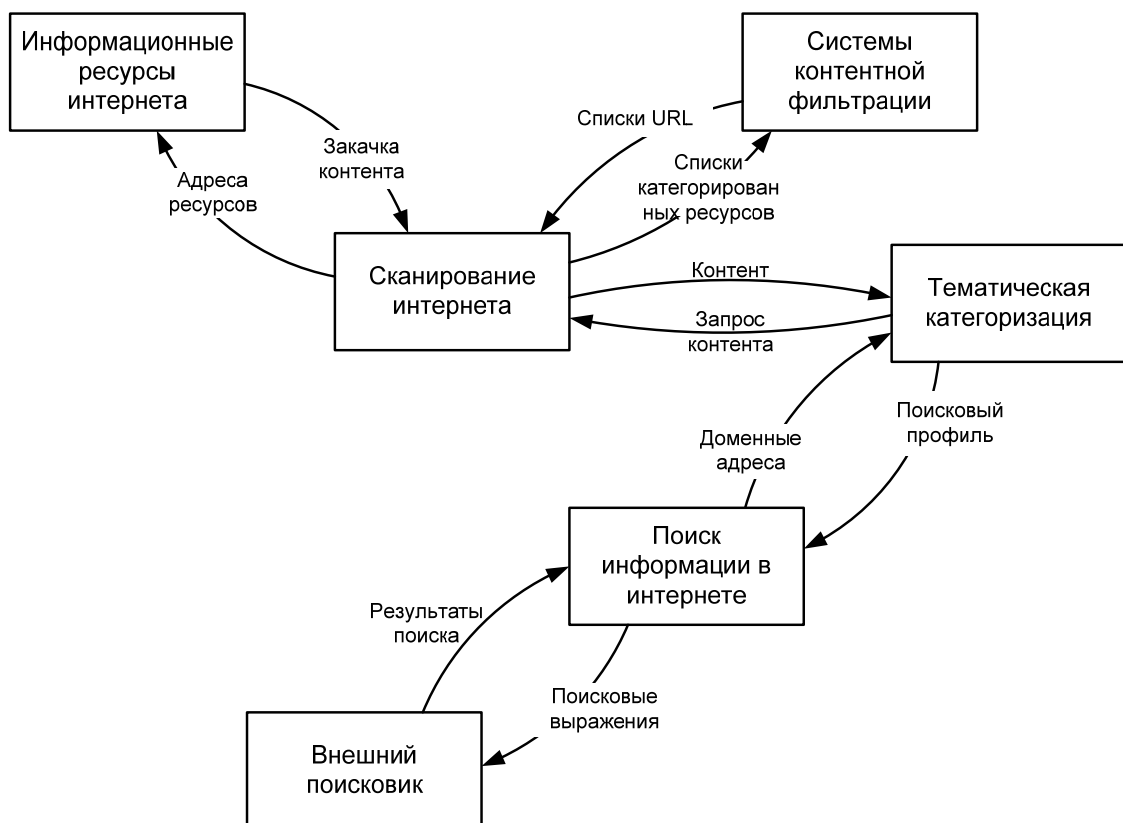


Рис. 3. Сбор и обработка данных интернет-ресурсов.

Блок поиска информации в интернете получает от блока тематической категоризации поисковые профили и преобразует их в поисковые выражения для внешних поис-

ковиков. Внешний поисковик передает результаты поиска обратно, далее выделяются доменные адреса информационных ресурсов и передаются блоку тематической категоризации.

Блок тематической категоризации запрашивает контент информационных ресурсов, адреса которых были добавлены в базу, но еще не были категоризованы, или те, для которых пришло время повторной категоризации.

Блок сканирования Интернета получает от систем контентной фильтрации списки доменных адресов для категоризации и передает обратно списки категоризованных ресурсов.

Блок сканирования Интернета по полученным адресам скачивает контент информационных ресурсов интернета и возвращает этот контент для проведения категоризации.

Для решения задач обработки информации интернет-ресурсов используются следующие два алгоритма:

1. поступление и обработка нового ресурса;
2. актуализация существующих интернет-ресурсов.

Первый алгоритм работает по следующему сценарию: находится новый ресурс, проверяется, есть ли он в каталоге, после чего проводится обход сайта (загрузка некоторого количества страниц), анализ этих страниц и присвоение категории сайту. Если этого количества страниц недостаточно для присвоения категории, проводится расширенный обход сайта (загружается большее количество страниц).

Актуализация существующих интернет-ресурсов проходит следующим образом: проводится новый обход сайта, проверяется, изменилась ли страница с момента последнего обхода, если изменилась, то проводится классификация загруженных страниц, если их достаточно, проводится классификация сайта, одновременно с этим уточняется принадлежность категориям.

Основная структура разрабатываемой системы тематической категоризации интернет-ресурсов и взаимодействие ее подсистем выглядит следующим образом (рис. 4).

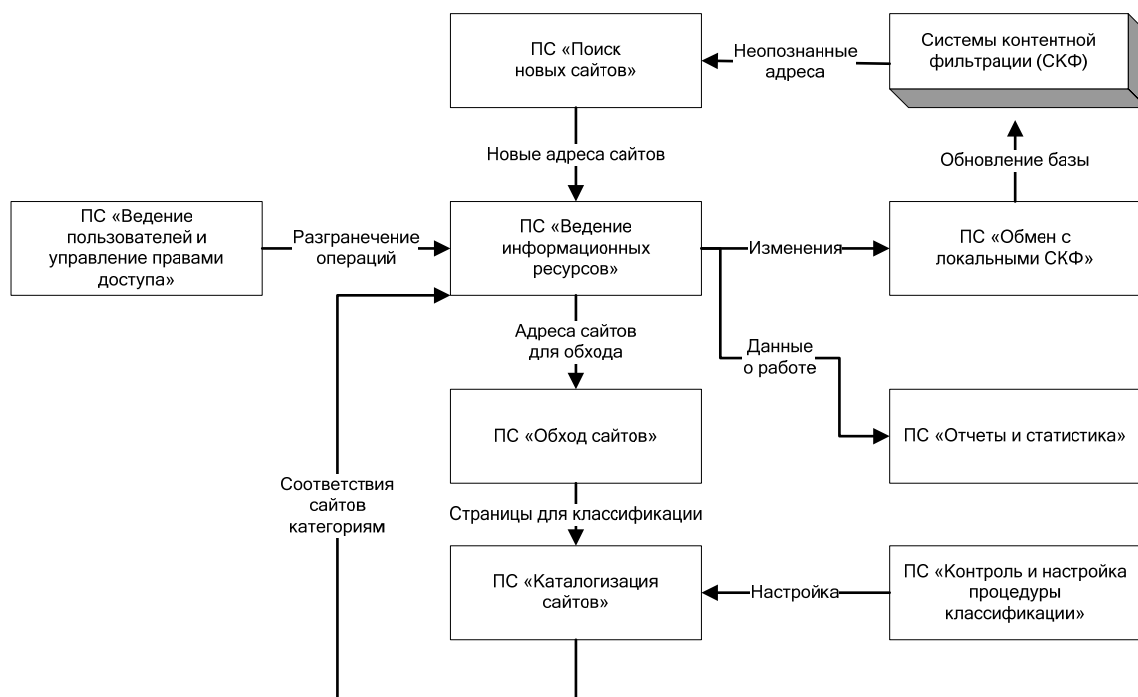


Рис. 4. Структура системы и взаимодействие ее подсистем

Подсистема «Поиск новых сайтов» предназначена для поиска новых интернет-ресурсов. Результатом ее деятельности является набор новых адресов сайтов, пополняющих базу тематической категоризации. На этом этапе информация о ресурсе является базовой, т.е. не содержит данных о категориях, к которым отнесен ресурс.

Далее вся информация о новых ресурсах поступает в подсистему «Ведение информационных ресурсов», где в результате классификации сайты получают соответствующие категории.

После этого в работу включается подсистема «Обход сайтов», в рамках которой осуществляется обход сайта и получение набора страниц для анализа тематики.

Следующей является подсистема каталогизации сайтов, которая анализирует тексты страниц, составляет их профиль и на основании этого принимает решение об отнесении сайта к той или иной тематической категории.

Процессом, контролирующим качество классификации, управляет подсистема «Контроль и настройка процедуры классификации».

Дополнительно ведется специализированный журнал отслеживания изменений о сайтах и категориях, который используется подсистемой «Обмен с локальными системами контентной фильтрации (СКФ)» для обновления данных в базах СКФ и получения от них новых, неизвестных адресов для анализа.

Подсистема «Ведение пользователей и управление правами доступа» позволяет использовать систему в многопользовательском режиме, с разграничением прав доступа между пользователями и контролем действий, выполняемых пользователем.

Подсистема «Отчеты и статистика» собирает информацию от всех модулей и предоставляет ее для анализа.

Разрабатываемые средства тематической категоризации интернет-ресурсов позволяют:

- составлять тематический каталог интернет-ресурсов за счет поиска веб-сайтов и поступления их от локальных систем контентной фильтрации при посещении пользователями веб-страниц;
- обеспечивать высокую точность категоризации интернет-ресурсов за счет составления тематических профилей при описании категорий;
- проводить актуализацию существующих интернет-ресурсов, т.е. заново осуществлять обход сайта, проверять изменения страниц, уточнять принадлежность сайта категориям;
- осуществлять обмен данными с локальными системами контентной фильтрации для защиты от нежелательного контента.

Литература

1. Фонд «Общественное мнение». <<http://www.fom.ru/>>.
2. Абсалямов А. Борьба с киберслэкингом // Windows 2000 Magazine, 2000. №3.
3. Плешко В.В., Ермаков А.Е., Голенков В.П. RCO на РОМИП 2004 / Российский семинар по оценке методов информационного поиска (РОМИП 2004). Пушкино, 2004. С. 43–61.
4. Плешко В.В., Ермаков А.Е., Митюхин В.А. RCO на РОМИП 2003: отчет об участии в семинаре по оценке методов информационного поиска / Труды Первого российского семинара по оценке методов информационного поиска. Под ред. И.С. Некрестьянова – СПб: НИИ химии СПбГУ, 2003. С. 42–51.
5. Поляков И.Е. Опыт создания системы фильтрации агрессивного web-контента / Труды XII Всероссийской научно-методической конференции «Телематика 2005», 6–9 июня 2005. СПб: СПбГУ ИТМО, 2005.

6. Sebastiani F. Machine Learning in Automated Text Categorization, <<http://nmis.isti.cnr.it/sebastiani/>>.
7. Некрестьянов И.С., Павлова Е.Ю. Обнаружение структурного подобия HTML-документов. / Труды Четвертой всероссийской конференции RCDL'2002. Дубна, Россия, 2002. С. 38–54.
8. Ziv Bar-Yossef, Sridhar Rajagopalan Template Detection via Data Mining and its Applications // Proceedings of WWW2002, May 7-11, 2002, Honolulu, Hawaii, USA.
9. Gupta S., Kaiser G., Grimm P., Chiang M., Starren J. Automating Content Extraction of HTML Documents // World Wide Web Journal. January 2005.
10. Russian Context Optimizer. Технологии анализа и поиска текстовой информации. <<http://www.rco.ru/>>.
11. Поляков П.Ю., Плешко В.В. RCO на РОМИП 2006 / Труды Четвертого российского семинара по оценке методов информационного поиска. СПб: НИИ химии СПбГУ, 2003. С. 72–79.

НЕКОТОРЫЕ ВОПРОСЫ ОТКРЫТОЙ АРХИТЕКТУРЫ ВЕРТИКАЛЬНЫХ ОБРАЗОВАТЕЛЬНЫХ ПОРТАЛОВ

А.В. Силаев

В процессе дальнейшего развития и совершенствования системы федеральных образовательных Интернет-порталов (СФОИП) наиболее перспективной является задача информационной интеграции вертикальных образовательных порталов (ВОП) с различными образовательными порталами/сайтами и горизонтальным порталом «Российское образование». Использование принципов открытой архитектуры для ВОП открывает ряд преимуществ, таких как получение доступа к сфере образовательных ресурсов посредством использования поисковых машин глобального поиска, унификации и распределению сервисов ВОП.

В настоящее время в русскоязычном сегменте Интернет функционирует и развивается территориально распределенная система федеральных образовательных Интернет-порталов (СФОИП). СФОИП состоит из центрального горизонтального портала «Российское образование» и совокупности вертикальных образовательных порталов (ВОП). Вертикальные образовательные порталы – широко востребованный продукт на российском образовательном рынке. Вертикальные образовательные порталы являются точкой входа в систему образовательных порталов и ориентированы на максимально широкий охват конкретной образовательной тематики или сферы деятельности. В процессе дальнейшего развития порталного обустройства наиболее перспективной является постановка задачи создания единого федерального распределенного образовательного портала и обеспечение его информационной интеграции с другими образовательными порталами и сайтами [1].

В силу своей специфики и предпочтений разработчиков большинство вертикальных порталов реализовано на разных технологических базах. Отличаются также их отдельные функциональные модули, технология наполнения и подсистемы поиска. Однако следует отметить и существование пробелов, заключающихся в недостаточной интеграции сервисов порталов и, в частности, межпортальных поисковых механизмов. Поиск ведется по документам и описаниям ресурсов, содержащимся только в базе данных конкретного портала. При этом в настоящее время такие ресурсы вручную или полуавтоматически добавляются в базу данных или в каталог образовательных ресурсов администраторами порталов. Поэтому, на взгляд автора, существует потребность в выработке решения, позволяющего с минимальными затратами интегрировать между собой порталы, построенные на разных технологических платформах и при этом наладить эффективный механизм поиска и обмена ресурсами между ними.

Типовая схема взаимодействия пользователя с сервисом вертикального портала представлена на рис. 1. Пользователь взаимодействует через Интернет с сервисом ВОП. Сервис представляет собой модуль или набор модулей – специальных внутренних подпрограмм, которые взаимодействуют через программное ядро с базой данных портала. Модули используют для генерации HTML-страниц, которые увидит пользователь, шаблоны, классы и обработчики. Шаблон – HTML-форма, которая будет заполнена обработчиком на основании полученных из БД портала данных. Класс представляет собой библиотеку команд конкретного сервиса для работы с программным ядром системы управления порталом и базой данных. Класс и обработчик предназначены для работы только с той системой управления порталом, для которой они предназначены.

При такой организации возможности поиска ограничиваются базой данных конкретного ВОП и поискового модуля, который «зашит» в систему управления порталом. Отсутствует возможность полнотекстового поиска по тематике портала в сети Интернет.

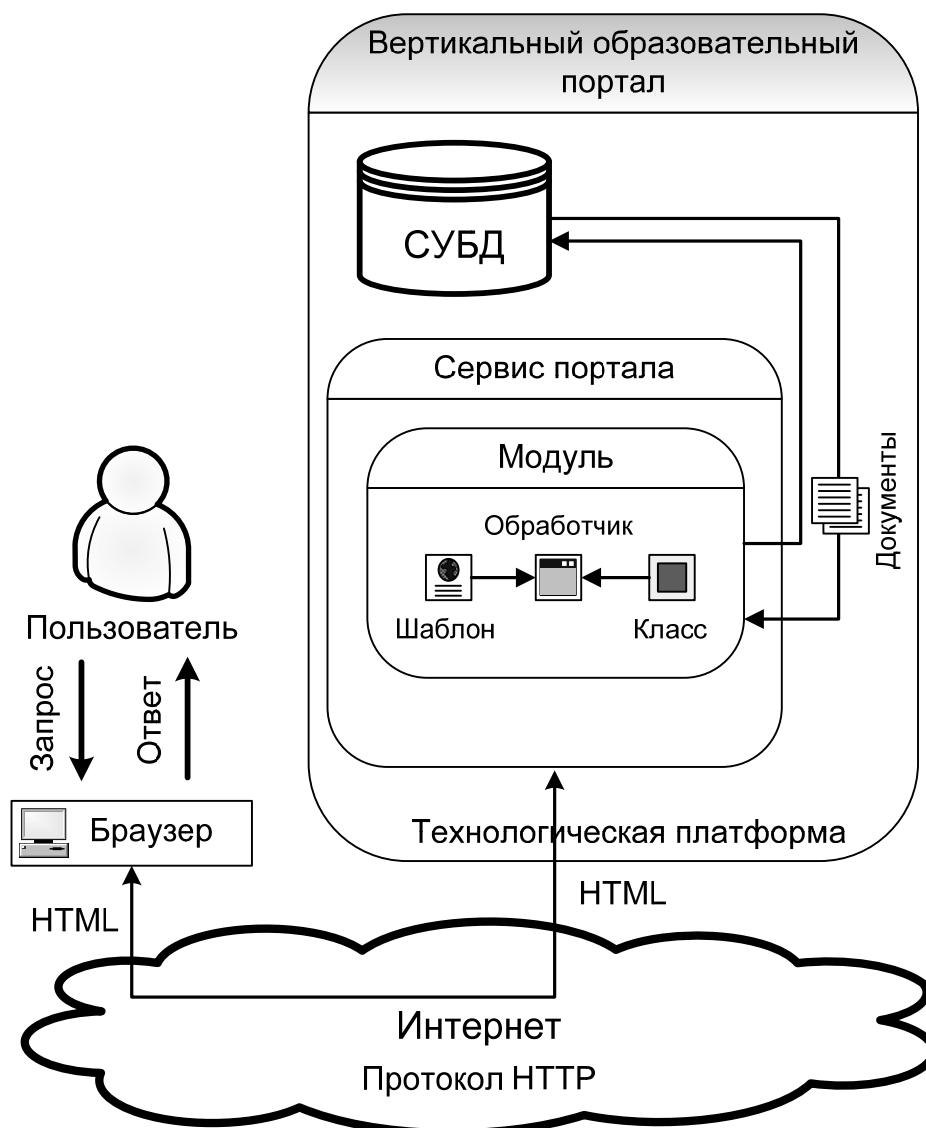


Рис. 1. Типовая схема взаимодействия пользователя с сервисом ВОП

Пути к решению поставленной задачи видятся автором в реализации концепции «открытой архитектуры ВОП». Под *открытой архитектурой вертикальных порталов* будем понимать архитектуру, на которую опубликованы спецификации, позволяющие другим (сторонним) производителям разрабатывать дополнительные или совершенствовать имеющиеся службы и сервисы к порталам с такой архитектурой, а также позволяющую использовать в качестве своих служб и сервисов функции других (сторонних) информационных систем по открытым спецификациям. При этом должны соблюдаться принципы модульности и масштабируемости для вертикальных порталов.

По этому пути в настоящее время идут крупнейшие зарубежные и отечественные коммерческие организации, такие как Microsoft, Oracle, IBM. Они переходят на порталы нового поколения, в основе которых лежит сервис ориентированная архитектура. Это, по мнению автора, говорит об актуальности исследования.

Открытая архитектура позволяет работать с удаленными сервисами и базами данных, интегрированными в составе других систем (рис. 2). Это достигается инициализацией соединения через сокет. Данные между сервисами передаются в формате XML. Модуль вертикального образовательного портала в этом случае состоит только из обработчика и шаблона. Обработчик ВОП взаимодействует с обработчиком информационной системы, сервисом которой требуется воспользоваться. Благодаря такому подходу

появляется возможность получать данные из СУБД сторонней системы непосредственно в вертикальный образовательный портал и передавать их пользователю. Преимущества такого подхода заключаются в независимости сервиса от технологической платформы портала.

Концепция открытой архитектуры позволяет осуществлять автоматическое взаимодействие сервисов систем, построенных на различных платформах, по общедоступным каналам связи. С ее помощью можно перенести отдельные сервисы на другие технологические платформы. При этом достаточно внести изменения в структуру одного сервиса, с которым работает вся система порталов, нет необходимости перепрограммировать сервис в каждом.

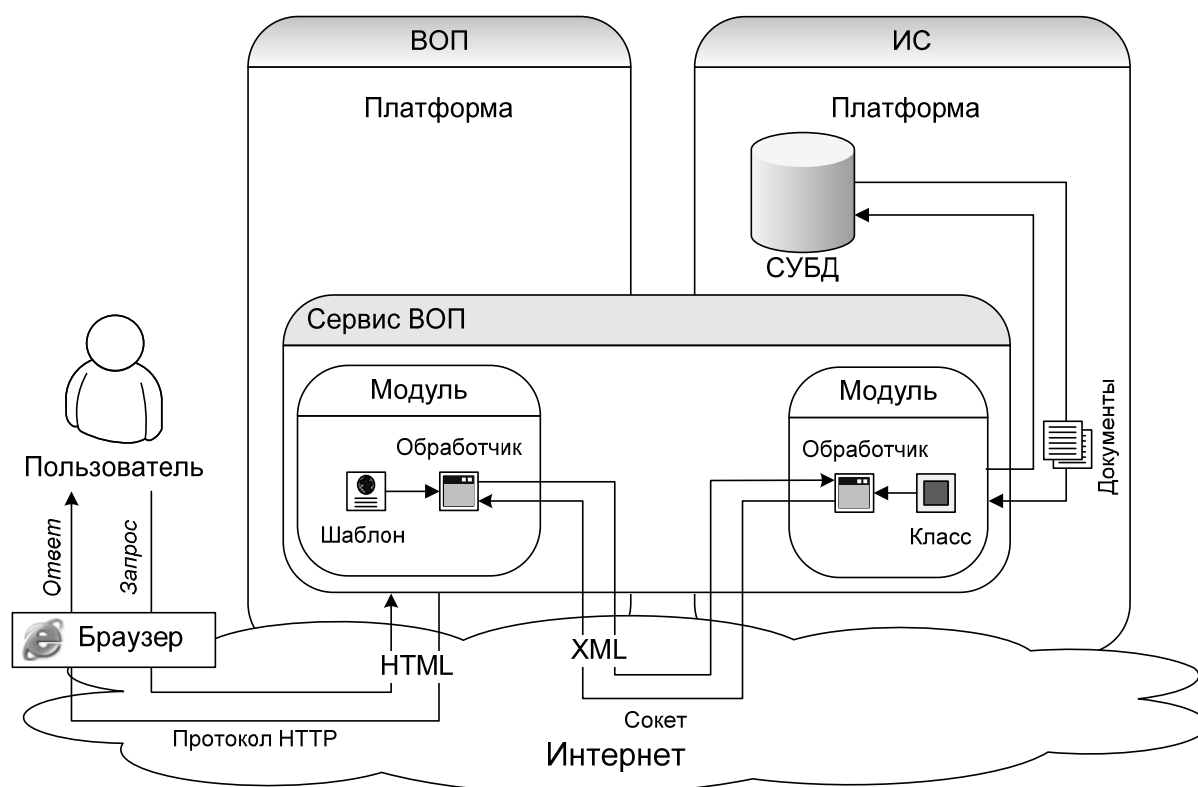


Рис. 2. Модель открытой архитектуры вертикальных образовательных порталов

С использованием принципов открытой архитектуры автором разработана модель интеграции поисковой машины yandex.ru непосредственно в интерфейс вертикального образовательного портала. При этом пользователь использует поисковые достижения Яндекса, находясь на вертикальном портале. Благодаря такой реализации пользователь получает дополнительные возможности, такие, как поиск информации по всей отрасли образования или по направлению деятельности вертикального портала. Для получения более широкой информации, которая не содержится на вертикальном портале, пользователю не приходится вручную разыскивать интересные ресурсы, что является достаточно утомительной процедурой. База данных, содержащая наиболее популярные и весомые, уже будет передана поисковой машине в запросе. Технологически это реализуется при помощи импорта/экспорта данных в открытом формате XML и выглядит следующим образом:

- XML-импорт фрагмента каталога «Российское образование», относящегося к сфере деятельности ВОП, во внутренний каталог ресурсов ВОП;
- выделение основных адресов сайтов, расположенных в каталоге;

- передача списка адресов поисковой машине yandex.ru с помощью технологии «Yandex XML».

Реализация поисковой системы ВОП с использованием открытой архитектуры и Яндекс.XML дает возможность поиска информации только по сайтам, имеющим прямое отношение к сфере деятельности портала.

На рис. 3 представлен алгоритм взаимодействия ВОП с поисковой системой. Запрос пользователя попадает в подсистему поиска, которая взаимодействует с базой данных (БД) ВОП. Если ответ (или группа ответов) имеется в БД ВОП, он выдается пользователю. Если нет – запрос конвертируется в XML-формат и отправляется поисковой системе. В свою очередь, он обрабатывается сервисом поисковой системы, который взаимодействует с внутренней БД. После этого результат конвертируется в XML-формат и возвращается ВОП. Помимо самого запроса пользователя, в поисковую систему можно передать тематику поиска или группу сайтов, на которых производить поиск. Ответом поисковой системы будут результаты поиска по группе сайтов сферы деятельности, которую охватывает ВОП. При этом пользователь получит их, находясь на вертикальном портале, не затрачивая дополнительное время на анализ Интернет.

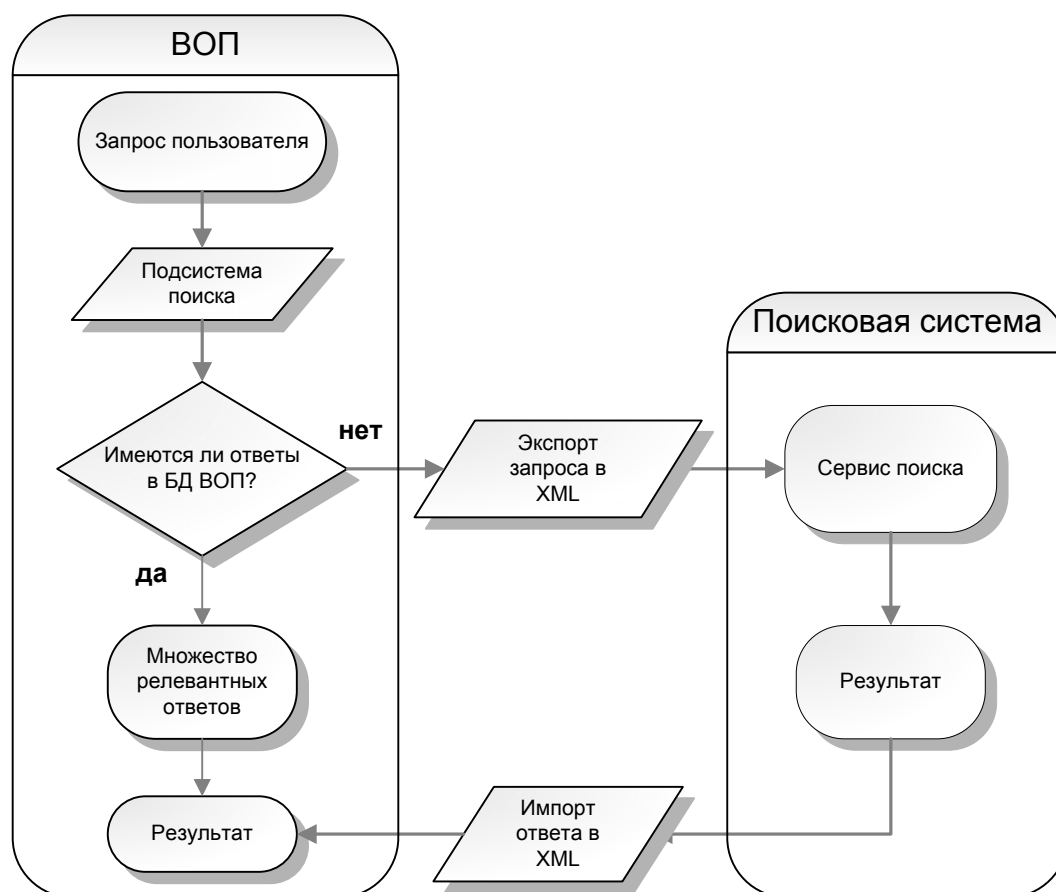


Рис. 3. Алгоритм взаимодействия ВОП с поисковой системой

Преимущества разработанного подхода можно рассмотреть с позиций теории множеств (рис. 4). Рассмотрим поисковое множество всех документов A . В нем имеется множество R релевантных запросу документов, а также множество S документов, выданных поисковой машиной. Идеальным случаем является совпадение этих двух множеств, но на практике этого никогда не происходит. Для оценки эффективности поиска ИПС используют коэффициенты полноты и точности. Коэффициент полноты –

отношение выданных релевантных документов к общему количеству релевантных документов в поисковом массиве:

$$K_R = \frac{R \cap S}{R}.$$

Коэффициент точности – отношение выданных релевантных документов к общему количеству выданных:

$$K_P = \frac{R \cap S}{S}.$$

Но эти коэффициенты не учитывают уместность результата.

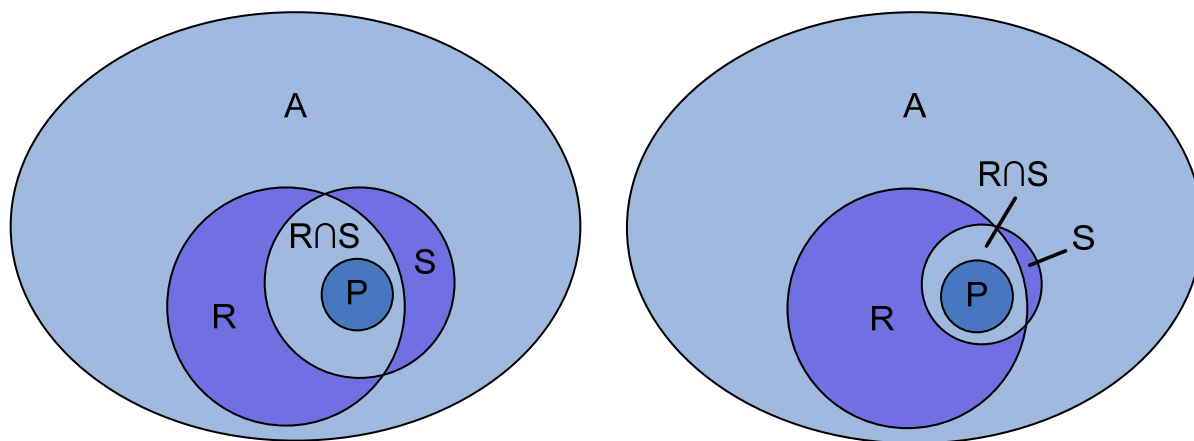


Рис. 4. Критерии оценки эффективности поиска по вертикальным образовательным порталам

Автором введен коэффициент информационной потребности, который определяется как отношение пертинентных (соответствующих информационной потребности) документов к общему количеству выданных ИПС релевантных документов:

$$K_I = \frac{P}{R \cap S}.$$

Коэффициент учитывает уместность результатов поиска и измеряется от нуля до единицы: нуль – полное несоответствие; единица – идеальный случай, когда все найденные ИПС документы полезны пользователю. Благодаря использованию таких оценок удалось спрогнозировать зауженную область поиска по группе вертикальных образовательных порталов. Было проведено экспериментальное исследование, которое показало применимость подхода. При использовании зауженной области поиска (рис. 4, справа) сокращается множество S документов, выданных поисковой машиной, и увеличивается коэффициент информационной потребности. При поддержании баланса между коэффициентом информационной потребности и энтропией поисковых ответов системы можно добиться наиболее пертинентных результатов работы вертикального образовательного портала, что является направлением дальнейшего исследования.

Применение открытой архитектуры для ВОП позволяет равномерно распределить обеспечение пользователей портала как первичными, так и вторичными (не представленными на портале) информационными ресурсами, унифицировать технологию обмена и регулярного обновления метаописаний между образовательными порталами, реализованными на разных платформах, использовать различные модули (сервисы) других порталов непосредственно из своего интерфейса. Использование принципа открытой архитектуры позволяет удерживать пользователя на портале, при этом он будет пользоваться общим образовательным каталогом всей системы порталов для получения информации. Использование открытой архитектуры как средства обмена данными с

внешними поисковыми системами позволяет подойти к решению задачи сокращения количества порталов. Это возможно за счет их интеграции в близкие по тематике. Для этого определяются критерии, влияющие на эффективность получения результата пользователем ВОП, такие как популярность (посещаемость) ресурса, его вес, обновляемость (свежесть информации). На их основании проводится оценка популярности вторичных образовательных ресурсов, упрощается фильтрация малозначимых и неиспользуемых ресурсов.

Наряду с перечисленными выше достоинствами открытая архитектура позволяет также упростить создание и поддержку коммуникационных площадок коллективного и индивидуального общения профессиональных групп разработчиков и пользователей образовательных Интернет-ресурсов.

Немаловажным может оказаться и то обстоятельство, что с помощью различных технических реализаций открытой архитектуры порталов можно добиться единой интеграции служб и сервисов системы порталов в целом. К примеру, станет более доступным использование единых новостных лент, систем рассылки, систем тестирования, голосований и систем рекламы, а также упростится использование единого каталога образовательных ресурсов.

Литература

1. Иванников А.Д., Булгаков М.В., Гридина Е.Г. «Современное состояние и перспективы развития системы федеральных образовательных порталов» / Интернет-порталы: содержание и технологии: Сб. науч. ст. Вып. 3 / Редкол.: А.Н. Тихонов (пред.) и др.; ГНИИ ИТТ «Информика». – М.: Просвещение, 2005. – 590 с.
2. Симонов А.В. «Динамика и перспективы развития образовательных Интернет-ресурсов» – Интернет-порталы: содержание и технологии: Сб. науч. ст. Вып. 3 / Редкол.: А.Н. Тихонов (пред.) и др.; ГНИИ ИТТ «Информика». – М.: Просвещение, 2005. – 590 с.
3. Иванников А.Д., Тихонов А.Н. «Основные положения концепции создания системы образовательных порталов» – Интернет-порталы: содержание и технологии: Сб. науч. ст. Вып. 1 / Редкол.: А.Н. Тихонов (пред.) и др.; ГНИИ ИТТ «Информика». – М.: Просвещение, 2003. – 720 с.
4. Бабинский А.З., Букатов А.А., Шапиро В.А., Шаройко О.В. Определение базовых сервисов, разработка методики наполнения и методов реализации образовательных порталов – Интернет-порталы: содержание и технологии. Сб. науч. ст. Вып.1 / Редкол.: Тихонов А.Н. (пред.) и др.; ГНИИ ИТТ «Информика». – М.: Просвещение, 2003. – 720 с.
5. Мордвинов В.А. «Аналитические средства и инженерные расчеты в практике проектирования ИС, порталов и картелей. Обеспечение качества сертифицируемых ПС в проектах (методическое пособие)» М., ОФАП Минобрнауки РФ, 03.06.04., №3637.
6. Соловьев С.Ю. Служба тематических толковых словарей Glossary Commander, Copyright 2000-2007 «Web-and-Press», <www.glossary.ru>.
7. Боркус В. Практическое построение SOA: борьба с мифами // PCWeek/RE. 2006. №48.
8. Силаев А.В. Применение синергетических подходов к исследованию информационных систем в образовании / Сборник тезисов докладов участников V региональной научно-практической конференции «Профессиональная ориентация и методика преподавания в системе школа–вуз». 22 апреля 2004 г. М.: МИРЭА, 2005. Т. I, с. 92–95.

**РАЗВИТИЕ НЕИНВАЗИВНЫХ МЕТОДОВ ОЦЕНКИ ВХОДНОЙ
ФУНКЦИИ ПО ПЭТ-ИЗОБРАЖЕНИЯМ****Е.Н. Терещенко**

В статье рассмотрена проблема развития неинвазивных методов оценки входной функции по ПЭТ-изображениям. Рассмотрен новый подход к данной проблеме, основанный на анализе трех областей исследования объекта для определения входной функции.

Введение

В сфере нарушений мозговой деятельности диагностика при анализе состояния пациентов до и после лечения методом позитронно-эмиссионной томографии (ПЭТ) является определяющей. ПЭТ позволяет изучить и проанализировать функциональное и биохимическое состояние головного мозга неинвазивным путем. Но одной лишь визуальной оценки ПЭТ-изображений недостаточно. Во-первых, цветовая шкала, отражающая на полученном изображении концентрацию глюкозы, не позволяет увидеть незначительные изменения в процессах, происходящих в изучаемой области головного мозга пациента, но являющимися очень важными для выбора дальнейшего хода лечения. Во-вторых, уровень накопленной в ткани и зарегистрированной во время сканирования активности является не абсолютной физиологической информацией об исследуемом процессе, выраженной в физиологических единицах (ммоль/100 г ткани/мин), а лишь его линейной оценкой, так как его значение зависит еще и от ряда технических причин – введенной активности, времени сканирования, текущей чувствительности датчиков камеры и т.д. Для расчета абсолютных значений применяются фармакокинетические модели, учитывающие динамику биохимического поведения РФП и так называемую входную функцию, т.е. кривую активности трейсера в плазме артериальной крови за время исследования. Знание абсолютных величин остается важным при оценке результатов медикаментозного лечения.

Описание метода

Метод ПЭТ – одна из немногих возможностей изучения не анатомии, а функционального состояния организма и биохимии *in vivo*. С технической точки зрения позитронно-эмиссионный томограф измеряет распределение радиоактивного трейсера (вещества, меченного ультракороткоживущими (УКЖ) позитрон-излучающими изотопами) в исследуемом объекте. Метод основан на использовании свойства неустойчивости ядер УКЖ изотопов, в которых количество протонов превышает количество нейтронов. При переходе ядра в устойчивое состояние оно излучает позитрон, свободный пробег которого заканчивается столкновением с электроном и их аннигиляцией. Аннигиляция сопровождается образованием двух γ -квантов с энергией 511 кЭв, разлетающихся под углом 180°, которые и регистрируются датчиками камеры [1].

Для исследования различных биохимических процессов требуются специально выбранные радиотрейсеры, или, иначе – радиофармпрепараты (РФП), поведение которых может быть аналитически осмыслено и для которых может быть построена адекватная математическая модель, связывающая доступные для измерения величины (такие как накопление трейсера или концентрация РФП в плазме артериальной крови) с исследуемыми параметрами [2].

Поскольку среди позитронных эмиттеров имеются изотопы основных элементов-органогенов (углерода, азота и кислорода), имеется принципиальная возможность использовать в качестве радиотрейсеров самые разнообразные биологически важные соединения – как содержащиеся в нормально функционирующем живом организме, так и ксенобиотики.

В связи со сказанным ПЭТ может быть использована для измерения региональных биохимических и физиологических параметров процессов, происходящих при участии химических соединений, меченных позитронными эмиттерами. В частности, при использовании метода ПЭТ при исследованиях живого мозга появляется возможность количественной оценки важнейших функциональных физиологических параметров – мозгового и коронарного кровотока (кровотока в сердечной мышце), метаболизма глюкозы и аминокислот, а также статуса рецепторов, ферментных систем и т.д. Данные о распределении радионуклида в обследуемом участке тела могут быть превращены с помощью соответствующей фармакокинетической модели в физиологическую или биохимическую информацию. Однако для расчета интересующих параметров недостаточно данных, получаемых при помощи позитронно-эмиссионного томографа, т.е. уровня накопленной активности в ткани. Для решения модельных уравнений требуется знание и так называемой входной функции, т.е. уровня активности в плазме крови в течение исследования.

Стандартный вид кривой входной функции представлен на рис. 1 и 2. На первом из них показано поведение кривой входной функции в первые 200 с после инъекции, а на втором — поведение кривой в последующее время, начиная с 1200-й с после инъекции. По оси абсцисс показано время в секундах, а по оси ординат – активность в беккерелях. Сплошная линия показывает теоретическую кривую, а прерывистая – экспериментальную [3].

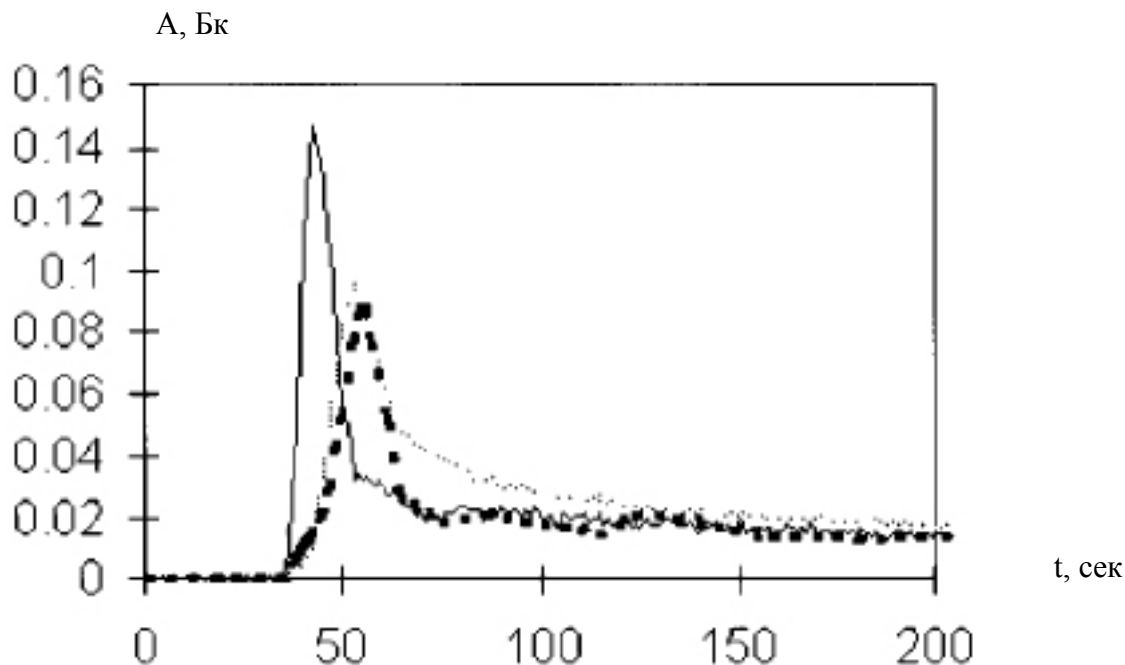


Рис. 1. Кривая входной функции в первые 200 сек после инъекции

Из рисунков видно, что пик активности фиксируется в первые секунды после инъекции, а в последующее время она убывает. По литературным данным от 85% до 95% РФП в первые минуты после инъекции проходит из сосудистого русла в ткань. Помимо этого, необходимо учитывать периоды полураспада, которые в среднем равны

2 мин, что через 10 мин приведет к полному распаду вещества. Из сказанного следует, что при любом поведении РФП (останется ли он в сосудах или полностью перейдет в ткань) справедливо утверждение, что через 25–30 мин показатель активности абсолютно точно будет равен нулю.

В настоящее время известны методы как инвазивного (взятие проб крови во время исследования и измерения активности независимыми методами), так и неинвазивного определения входной функции. Речь идет об определении входной функции путем измерения активности в артериальной крови и анализе ПЭТ-изображений.

Если рассматривать первый метод (метод определения входной функции путем измерения активности в артериальной крови), то, несмотря на то, что он дает возможность непрерывного измерения, данный метод имеет ряд существенных недостатков. Во-первых, это довольно болезненный метод, во-вторых, существует временная погрешность, которая возникает за счет сдвига времени прихода болюса (значение активности в артерии все равно будет отличаться от значения входной функции в головном мозге), в-третьих, возникает необходимость кросс-калибровки. Нельзя забывать и о человеческом факторе – погрешность в измерения может быть внесена и медицинскими работниками, осуществляющими забор крови из артерии.

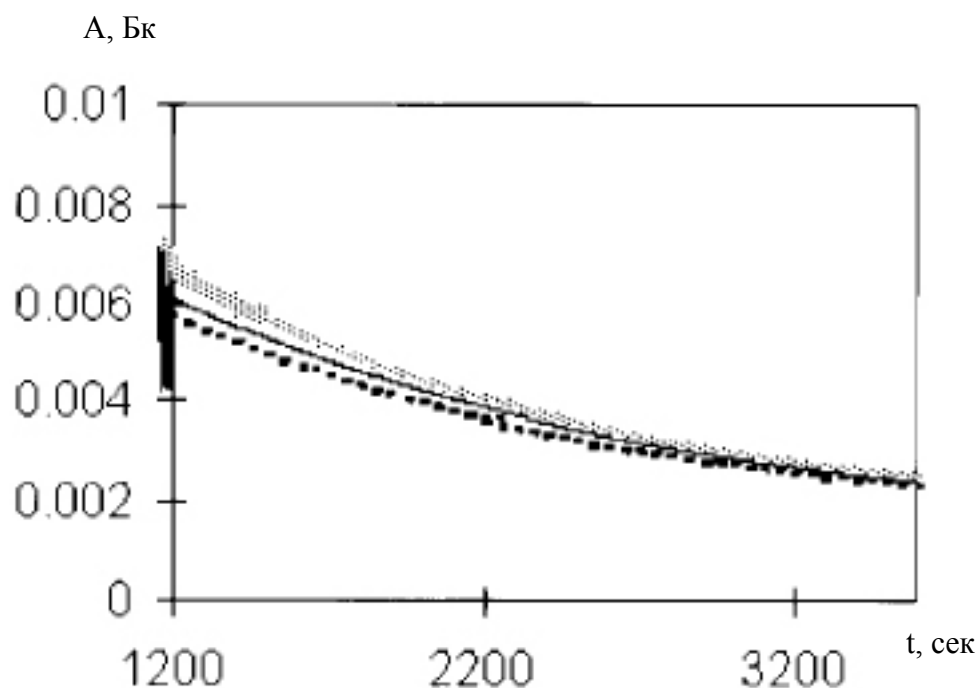


Рис. 2. Кривая входной функции в последующее время после инъекции, начиная с 1200 сек

Метод анализа входной функции по ПЭТ-изображению лишен вышеперечисленных недостатков. Единственным известным в настоящее время недостатком данного метода может являться недостаточное пространственное разрешение томографа.

В настоящее время идет разработка нового неинвазивного метода определения входной функции. Данный метод основан на рассмотрении трех областей интереса. В качестве первой области интереса мы будем рассматривать крупный сосуд головного мозга, затем область прилегающих к выбранному сосуду тканей головного мозга, в которые уходит со временем часть введенной пациенту активности. Третья область будет выбираться в той части головного мозга, о которой известно наверняка, что крупные

сосуды в ней не присутствуют. Во всех трех выбранных областях необходимо будет определить входную функцию [4].

В ходе работы предполагается найти уравнение входной функции, описывающей поведение РФП при данных условиях во всех трех выбранных областях интереса, разработать алгоритм определения входной функции и создать программный продукт, который бы отвечал всем предъявляемым к нему требованиям в процессе определения искомой входной функции. Результаты работы будут опробованы на имеющейся материальной базе лаборатории ПЭТ Института мозга человека РАН.

Заключение

Современные условия таковы, что к настоящему моменту необходим переход к абсолютным числам. Одной цветовой шкалы недостаточно, чтобы уловить малейшие изменения, которые могут оказать большое влияние на дальнейшее состояние пациента. В то же время использование ПЭТ в фармакологической промышленности и знание абсолютных величин при этом в 1,5 раза ускоряет процесс внедрения препарата в практику. Большое значение имеет знание входной функции, инвазивное определение которой, как показала многолетняя практика, является трудоемким для персонала, а также тяжело переносится пациентами и несет в себе множество погрешностей. Поиск неинвазивных методов определения входной функции ведется достаточно давно, их актуальность доказана на практике.

Литература

1. Терещенко Е.Н. ПЭТ и один из подходов к повышению точности анализа изображения / Междунар. конф. «Приборостроение в экологии и безопасности человека». СПб, 10–12 ноября 2004 г. Труды. С. 102–103.
2. Messa C, Goodman WG, Hoh CK, et al. Bone metabolic activity measured with positron emission tomography and 18 F-fluoride in renal osteodystrophy: correlation with bone histo-morphometry // J. Clin Endocrinol Metab. 1993; 77: 949–955.
3. Image-Derived Input Function for [11C]Flumazenil Kinetic Analysis in Human Brain. Sandra M. Sanabria-Bohórquez, PhD1, Alex Maes, Patrick Dupont, Guy Bormans, Tjibbe de Groot, Alexandre Coimbra, PhD1, WaiSi Eng, Tine Laethem, Inge De Lepeleire, Jay Gambale, Jose M. Vega, H. Donald Burns, Molecular Imaging and Biology, Vol. 5, No. 3, 72–78. 2003
4. Yu-Hua Fang, Tsair Kao, Ren-Shyan Liu, Liang-Chih Wu, Estimating the input function non-invasively for FDG-PET quantification with multiple linear regression analysis: simulation and verification with in vivo data. Springer-Verlag 2004

ТЕОРЕТИКО-ГРУППОВЫЕ МЕТОДЫ ФОРМИРОВАНИЯ ШКАЛ КОРРЕКЦИИ ПРОСТРАНСТВЕННЫХ ИСКАЖЕНИЙ ИЗОБРАЖЕНИЙ В ТОМОГРАФИИ

Н.Д. Скалецкая

Основной целью работы является определение параметров групповых преобразований, отражающих изменение положения исследуемого объекта в пространстве относительно эталонного, а также теоретическое обоснование методики коррекции пространственных искажений изображений объектов в томографии.

Введение

Разные томографические изображения одного и того же объекта могут существенно отличаться друг от друга. Причинами этих отличий являются искажения, связанные с техническими погрешностями системы получения и передачи изображения, яркостные искажения, а также пространственные искажения, связанные с изменением положения в пространстве исследуемого объекта.

Процесс обработки изображений объектов включает в себя формирование в устройстве регистрации некоторого внутреннего представления изображения, например, формирование в памяти компьютера матрицы значений яркости элементов изображения, и передачу этой информации системе обработки.

В математической форме процесс формирования изображения можно описать как отображение пространства R^3 в R^2 , т.е. $F: R^3 \rightarrow R^2$ [4]. Сцена может быть задана в пространстве R^3 как функция $\Psi(\varepsilon, \rho, \theta)$, тогда изображение есть результат отображения

$$f(x, y) = F[\Psi(\varepsilon, \rho, \theta)],$$

где x, y – координаты пространства R^2 , $\varepsilon, \rho, \theta$ – координаты пространства R^3 , F – отображение. Значение функции $f(x, y)$ принято называть яркостью изображения в точке (x, y) плоской декартовой системы координат.

В силу того, что в процесс получения изображения на практике всегда вносятся искажения, в систему обработки поступает искаженное изображение $f'(x', y') = A[f(x, y)]$, где A – оператор, описывающий искажения, вносимые в изображение $f(x, y)$, которое можно назвать эталонным. Таким образом, задача устранения искажений сводится к нахождению обратного оператора A^{-1} и применению его к искаженному изображению.

Все искажения можно условно разделить на пространственные, описываемые оператором A_1 , и яркостные, описываемые оператором A_2 [2]. Тогда

$$A = A_1 \cdot A_2,$$

причем порядок следования операторов определяется спецификой процесса получения изображения, так как оператор A_2 в общем случае зависит от координат (x, y) .

Под пространственными искажениями можно понимать некоторые отклонения при отображении точек пространства R^3 в R^2 , когда результат проецирования точки $(\varepsilon, \rho, \theta)$ в $R^2(x', y')$ не совпадает с точкой $(x, y) = F(\varepsilon, \rho, \theta)$; под яркостными искажениями – некоторые отклонения при отображении сцены при отсутствии пространственных искажений, когда результат проецирования $f'(x', y')$ отличается от $f(x, y)$ при $x = x'$, $y = y'$.

Особенностью коррекции пространственных искажений является необходимость определения оператора A_1^{-1} , связывающего координаты точек эталонного и искаженного изображений и не зависящего от функции яркости, т.е. $X' = A_1(X)$, где $X' = (x', y')$, $X = (x, y)$ – векторы координат точек соответственно искаженного и эталонного изображе-

ний. Предполагается $f'(x', y') = f(x, y)$, что и позволяет провести коррекцию пространственных искажений.

Постановка задачи

Задачу определения положения исследуемого объекта в пространстве можно сформулировать следующим образом [5]. Известны начальное (эталонное) положение объекта в трехмерном пространстве, его изображение в этом положении и изображение объекта после изменения его положения в пространстве. Требуется определить параметры, описывающие изменение положения объекта относительно эталонного.

Движение тела в пространстве можно описать шестью параметрами: x_0, y_0, z_0 – смещение некоторой точки O тела соответственно по осям X, Y, Z неподвижной системы координат; α, β, γ – углы поворота тела вокруг осей подвижной системы координат X_0, Y_0, Z_0 , связанной с началом координат точкой O [1]. В матричной форме изменение положения точки тела можно представить как

$$X' = A_3(\gamma)A_2(\beta)A_1(\alpha)X + X_0,$$

где $X = (x, y, z)^T$, $X' = (x', y', z')^T$ – вектора-столбцы координат некоторой точки тела соответственно до и после изменения положения в пространстве, $X_0 = (x_0, y_0, z_0)$ – вектор-столбец смещения точки O ,

$$A_1(\alpha) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) \\ 0 & \sin(\alpha) & \cos(\alpha) \end{pmatrix}, A_2(\beta) = \begin{pmatrix} \cos(\beta) & 0 & \sin(\beta) \\ 0 & 1 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) \end{pmatrix},$$

$$A_3(\gamma) = \begin{pmatrix} \cos(\gamma) & -\sin(\gamma) & 0 \\ \sin(\gamma) & \cos(\gamma) & 0 \\ 0 & 0 & 1 \end{pmatrix} \text{ – матрицы поворота соответственно вокруг осей } X_0, Y_0,$$

Z_0 . При параллельном проецировании координаты (x'', y'') точки изображения совпадают с координатами (x', y') соответствующей точки тела.

При центральном проецировании выполняется соотношение

$$\frac{x''}{f} = \frac{x'}{z'}, \frac{y''}{f} = \frac{y'}{z'},$$

где f – фокусное расстояние устройства получения изображения.

Определенный интерес представляет зависимость параметров группового преобразования от значений углов поворота вокруг осей X, Y, Z . По известным значениям вектора X_0 , углам α, β, γ , фокусному расстоянию f и координатам характерных точек на эталонном изображении объекта можно определить параметры группового преобразования.

Методы исследования и полученные результаты

Ввиду того, что пространственные искажения изображений меняются плавно, они могут быть описаны некоторыми непрерывными групповыми преобразованиями. В качестве эффективного математического аппарата при решении задач обработки изображений исследуемых объектов, полученных с помощью магнитно-резонансных, позитронно-эмиссионных и рентгеновских компьютерных томографов, используются группы Ли [6].

Подвергнув искаженное изображение преобразованию элементом g_1^{-1} , можно обеспечить «сближение» искаженного и эталонных изображений. Продолжая процесс последовательной нейтрализации подгрупп, можно либо полностью устранить искажения (определить параметры положения объекта), либо упростить задачу и свести ее к известному решению. В качестве первых шагов обычно выбираются некоторые характеристики изобраа-

жения, такие, например, как центр формы или центр функции яркости фигуры, ее площадь и т.п. Последующие шаги обеспечиваются использованием характерных (так называемых «реперных») точек на изображении.

На рис.1 представлены МРТ-снимки фантома (шесть шприцев, заполненных водой и жестко скрепленных между собой), полученные на МР-томографе GE Signa Infinity 1.5 T.

Выберем на эталонном и искаженном изображениях реперные точки (наименее искаженные точки на соответствующих изображениях) и относительно них найдем параметры группового преобразования [6]. Предположим, что искаженное изображение фантома явилось следствием вращения исследуемого объекта относительно одной или нескольких координатных осей. Попробуем выявить зависимость параметров группового преобразования от углов поворота образца относительно осей вращения.

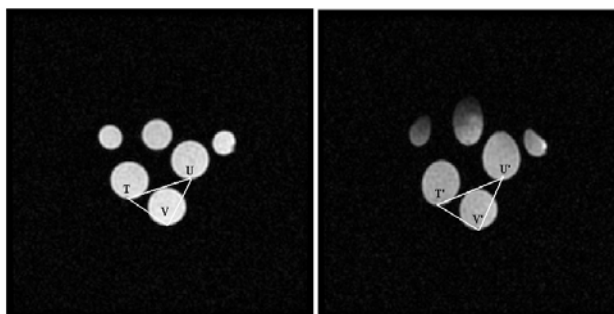


Рис. 1. Эталонное и искаженное изображения фантома. Т, U, V, Т', U', V' – реперные точки

Погрешность определения координат точек $\Delta = \pm 1$ пикс. = ± 0.6 мм.

Так как наше изображение было предварительно центрировано, то точка О совпадает с осями X, Y, Z неподвижной системы координат, т.е. $X_0 = 0$.

Координаты характерных точек Т, U, V известны [3]. Углы поворота возьмем равными $0^\circ, 30^\circ, 60^\circ, 90^\circ$.

Уравнения изменения положения точки тела в нашем случае имеют следующий вид:

- $X'_X = A_1(\alpha)X$ – изменение положения точки тела относительно оси X;
- $X'_Y = A_2(\beta)X$ – изменение положения точки тела относительно оси Y;
- $X'_{XY} = A_2(\beta)A_1(\alpha)X$ – изменение положения точки тела относительно осей X, Y;
- $X'_{XYZ} = A_3(\gamma)A_2(\beta)A_1(\alpha)X$ – изменение положения точки тела относительно осей X, Y, Z.

При расчете аффинных преобразований будут использоваться формула аффинной группы без сдвига:

$$\begin{cases} x' = a_1x + a_2y \\ y' = a_3x + a_4y \end{cases},$$

где x, y, x', y' – исходные и преобразованные аффинной группой координаты точки изображения, $a_1, \dots, a_4, b_1, b_2$ – параметры аффинной группы; и формула проективной группы:

$$\begin{cases} x' = \frac{a_1x + a_2y}{a_5x + a_6y + 1} \\ y' = \frac{a_3x + a_4y}{a_5x + a_6y + 1} \end{cases},$$

где x, y, x', y' – исходные и преобразованные проективной группой координаты точки изображения, $a_1, \dots, a_6, b_1, b_2$ – параметры проективной группы [5].

Пределы изменения и постоянные значения параметров в зависимости от вида проецирования и сочетания осей вращения приведены на рис. 2 и 3.

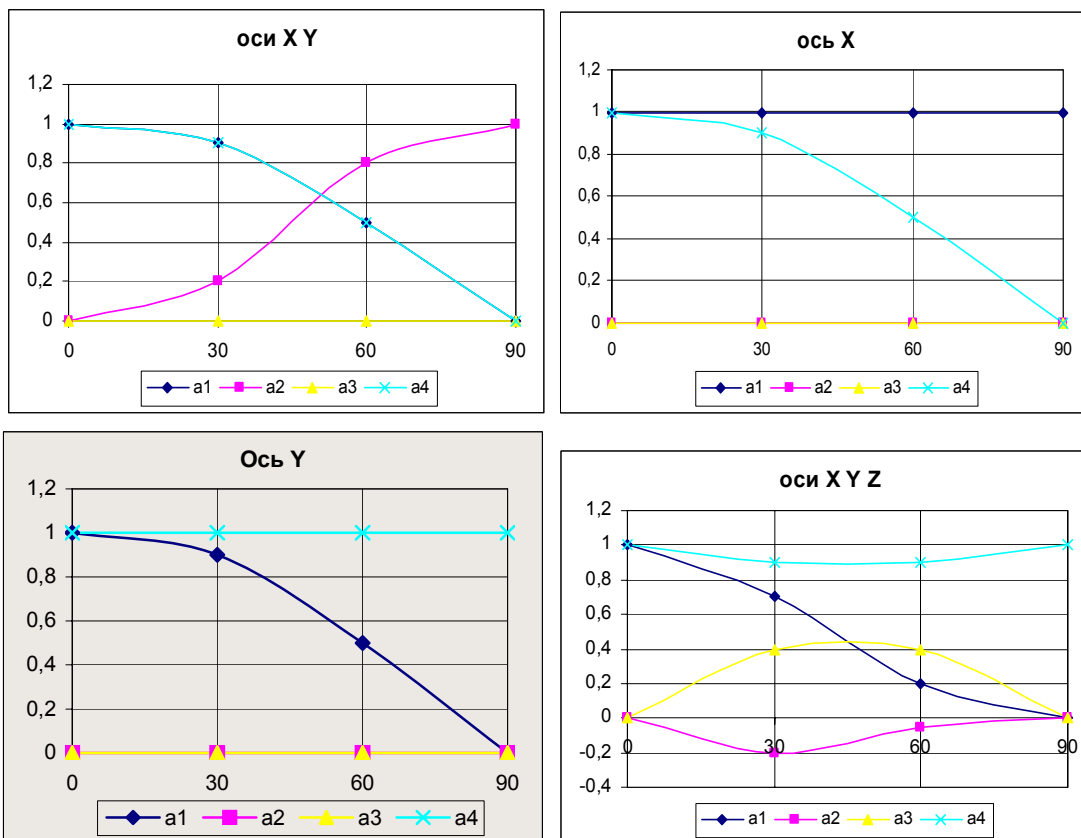


Рис. 2. Зависимость параметров группового преобразования от угла поворота объекта: преобразование изображения аффинной группой

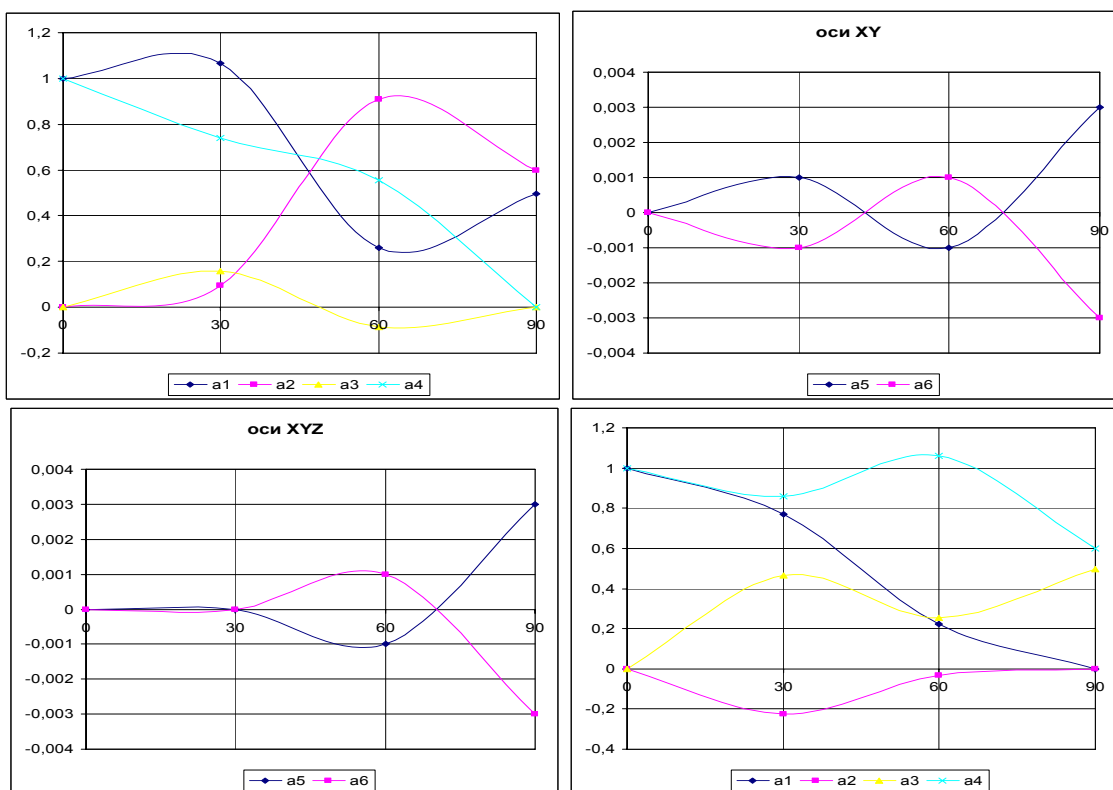


Рис. 3. Зависимость параметров группового преобразования от угла поворота объекта: преобразование изображения проективной группой

Заключение

Таким образом, смещение исследуемого объекта и, как следствие, его изображения по осям X и Y определяет параметры b_1 и b_2 , смещение по оси Z при параллельном проецировании вообще не влияет на изменение параметров группы, а при центральном проецировании определяет степень масштабных преобразований.

Из анализа рис. 2 и рис. 3 видно, что при отсутствии смещения и масштабного преобразования значения параметров $a_1, \dots, a_4$ меняются в пределах $-1, \dots, 1$, а значения a_5, a_6 не превышают 0.01.

Анализ групп преобразований и возможных изменений их параметров позволяет выделить следующие подходы к решению задачи коррекции пространственных искажений: разложение полной группы преобразования на подгруппы и их последовательная компенсация; нахождение параметров полной группы преобразований по координатам характерных (реперных) точек на изображении исследуемого объекта.

Пространственные искажения изображения, вызываемые погрешностями в работе системы получения и передачи изображения, в общем случае являются суммой некоторой нелинейной и случайной составляющих. В результате коррекции изображения эти искажения в той или иной степени компенсируются. Величину оставшегося после коррекции пространственного искажения можно считать одним из критериев качества коррекции.

Литература

1. Ерош И.Л. Элементы теории дискретных групп: Учебное пособие. СПб.: СПбГУАП, 1998. 40 с.
2. Иванов В.А., Марусина М.Я. Применение теории групп при решении задач реализации измерительных преобразований // Изв. вузов. Приборостроение. 2006. Т. 43, № 6. С. 36–39.
3. Марусина М.Я., Скалецкая Н.Д., Казначеева А.О. Коррекция пространственных искажений в томографии // Научное приборостроение. 2005. Т. 15. № 3. С. 77–82.
4. Смирнов А.Я. Цифровая обработка изображений. Л.: ГОИ, 1987. 50 с.
5. Файн В.С. Опознавание изображений: Основы непрерывно-групповой теории и ее применение. М.: Наука, 1970. 296 с.
6. Чеботарев И.Г. Теория групп Ли. М.: ГИТТЛ, 1940. 396 с.

СРАВНЕНИЕ ТРЕХ МЕТОДОВ СИНТЕЗА ОДНОРОДНОГО ПОЛЯ ПЕРМАНЕНТНОГО МАГНИТА В МР-ТОМОГРАФИИ

Ю.И. Неронов, В.С. Сизиков, Д.Ю. Соколов

Развивается метод эквивалентных витков определения оптимальной конфигурации осесимметричного постоянного магнита МР-томографа для формирования высокооднородного поля в зазоре магнита. При этом используется аналогия между поверхностями магнита и соответствующим набором круговых витков с током. Это позволяет при расчете поля магнита использовать формулы для магнитных полей витков с током. В методе с целью повышения однородности поля в полюсные наконечники магнита вводятся углубления сложного профиля. Параметры углублений определяются из условия минимума отклонения рассчитанного поля от однородного в некоторой рабочей области. Приведены результаты компьютерного моделирования. Произведено сравнение данного метода с методом скалярного потенциала и с методом диполей, а также с методом эквивалентного соленоида. Магнит может быть изготовлен как из сплава с высокой коэрцитивной силой, например, Nd-Fe-B, так и из мягкого железа с подпитывающей катушкой. Даны рекомендации по учету неоднородности намагниченности, по подавлению токов Фуко и по использованию данного метода в МР-томографии.

Введение

Одной из важнейших задач МР-томографии является задача формирования (синтеза) высокооднородного магнитного поля [1–6], так как только при высокой однородности поля ($\Delta H/H \sim 10^{-5} - 10^{-6} = 1 - 10 \text{ ppm}$) можно обеспечить получение томограмм с высоким разрешением [7, с. 51–52].

Во многих работах ([1–3, 8–16] и др.) рассматриваются *перманентные (постоянные) магниты* из ферромагнитных материалов. Преимущество постоянных магнитов перед сверхпроводящими и резистивными магнитами состоит в том, что постоянные магниты не требуют охлаждения жидким гелием или водой и питания электричеством. Однако они создают относительно низкие поля (0.1–0.4 Тл), но это не следует считать недостатком. В последние годы многие фирмы-производители МР-томографов (General Electric, Siemens, Toshiba и др.) стали уделять внимание выпуску среднеполюсных МР-томографов, поскольку повысились их технические и алгоритмические характеристики. Высокополюсные томографы (с полем выше ≈ 1 Тл) перестали иметь существенные преимущества для клинической диагностики перед средне- и даже низкополюсными томографами (кроме ЯМР-спектроскопии *in vivo* [17], где необходимы поля > 1.5 Тл).

Расчет конфигурации (формы, геометрии) постоянного магнита, при которой формируется высокооднородное магнитное поле, требует специального алгоритмического подхода. Это связано со следующими обстоятельствами. Существует ряд хорошо разработанных методов (способов) расчета магнитных полей, создаваемых витками с током различной конфигурации (изолированные витки, соленоиды, катушки с зазором и/или пазом и т.д.) [1–6, 18–20], и все эти методы основаны на законе Био–Савара–Лапласа [1–3, 18, 21]. Сложнее обстоят дела с расчетом магнитных полей, создаваемых постоянными магнитами, и это обусловлено тем, что: а) не существует реальных магнитных зарядов–монополей [21, с. 247, 611], поэтому используются диполи, что приводит к несколько более сложному закону, чем закон Био–Савара–Лапласа; б) в постоянном магните нужно учитывать такие явления, как намагничивание и размагничивание, гистерезис, неоднородность намагниченности, токи Фуко, неоднородность материала [1–3, 18]; в) имеет место более сложный математический аппарат расчета полей постоянных магнитов [1–3, 9–13, 22–25], чем витков с током. Тем не менее, можно отметить, что для расчета постоянных магнитов развиты следующие методы [3, с. 185]: метод отношений, метод размагничивающего фактора и метод эквивалентного соленоида, а также методы скалярного и векторного потенциалов и метод диполей [1–3, 9–13, 22–25]. Проанализируем некоторые из этих методов.

Метод скалярного потенциала. В работах [9, 11–13] для расчета магнитного поля, создаваемого постоянным магнитом сложной конфигурации (магнит с наконечниками и углублениями), используется *скалярный магнитный потенциал* u [1–3, 9, 11, 13, 21–26]. Для потенциала u в любой односвязной области пространства, не содержащей токов, справедливо *уравнение Лапласа*, которое в сферических координатах записывается в виде [9, 25]:

$$\frac{1}{r^2} \frac{\partial}{\partial r} \left(r^2 \frac{\partial u}{\partial r} \right) + \frac{1}{r^2 \sin \theta} \frac{\partial}{\partial \theta} \left(\sin \theta \frac{\partial u}{\partial \theta} \right) + \frac{1}{r^2 \sin^2 \theta} \frac{\partial^2 u}{\partial \varphi^2} = 0, \quad (1)$$

где r, θ, φ – сферические координаты, $u = u(r, \theta, \varphi)$ – скалярный магнитный потенциал. При этом должны быть добавлены еще граничные условия. *Напряженность магнитного поля* $\mathbf{H} = \mathbf{H}(r, \theta, \varphi)$ равна

$$\mathbf{H} = -\text{grad} u \equiv -\nabla u. \quad (2)$$

Общее решение уравнения (1) имеет вид [9, 11, 13, 25]:

$$u(r, \theta, \varphi) = \sum_{n=0}^{\infty} \sum_{m=0}^{\infty} \left(\frac{r}{r_0} \right)^n \left(A_n^{(m)} \cos m\varphi + B_n^{(m)} \sin m\varphi \right) P_n^m(\cos \theta), \quad r < r_0, \quad (3)$$

где

$$P_n^m(x) = (1-x^2)^{m/2} \frac{d^m P_n(x)}{dx^m},$$

r_0 – некоторая константа (обычно радиус сферы, внутри которой ищется решение (3)), $P_n(x)$ – полиномы Лежандра, $P_n^m(x)$ – присоединенные функции Лежандра, $A_n^{(m)}, B_n^{(m)}$ – коэффициенты, определяемые из граничных условий. Если, например, для заданной конфигурации магнита известны (измерены магнитометром) значения потенциала u на некоторой поверхности в ряде ее точек, то, подставляя в (3) его экспериментальные значения $u_i, i = 1, 2, \dots, K$, получим систему K уравнений, решая которую, найдем некоторое конечное число коэффициентов $A_n^{(m)}, B_n^{(m)}, n = 0, 1, \dots, N, m = 0, 1, \dots, M$. После этого потенциал u можно рассчитать во всех точках пространства (при $r < r_0$) по формуле (3) с конечными верхними пределами M и N суммирования по m и n .

Если на некоторой поверхности известны производные потенциала u , т.е. компоненты напряженности поля H_x, H_y, H_z , то нужно путем дифференцирования (3) найти

$$H_x = -\frac{1}{r} \frac{\partial u}{\partial \theta}, \quad H_y = -\frac{1}{r \sin \theta} \frac{\partial u}{\partial \varphi}, \quad H_z = -\frac{\partial u}{\partial r}, \quad (4)$$

а затем, подставив в (4) экспериментальные значения $(H_x)_i, (H_y)_i, (H_z)_i$, измеренные на поверхности, получим систему уравнений относительно $A_n^{(m)}, B_n^{(m)}$ [25].

Если же постоянный магнит и его магнитное поле симметричны относительно оси z , то потенциал u не зависит от φ и записывается в более простом виде [2, 13, 26]:

$$u(r, \theta) = \sum_{n=0}^{\infty} \left(\frac{r}{r_0} \right)^n Z_n P_n(\cos \theta), \quad r < r_0, \quad (5)$$

где Z_n – некоторые коэффициенты, определяемые из граничных условий.

Итак, первый метод требует экспериментальных значений потенциала u (или напряженностей H_x, H_y, H_z) в ряде точек на некоторой граничной поверхности. Затем путем решения системы уравнений определяются коэффициенты $A_n^{(m)}, B_n^{(m)}$ или Z_n . И, наконец, потенциал $u(r, \theta, \varphi)$ или $u(r, \theta)$ рассчитывается по формуле (3) или (5) с ко-

нечными верхними пределами суммирования по m и n . Если расчетное поле не удовлетворяет требованиям однородности, то вводят изменения в форму магнита и заново выполняют перечисленные операции, в том числе, экспериментальные измерения.

Видим, что данный метод является весьма трудоемким. Тем не менее, он позволяет решать задачу синтеза поля в самом общем виде, а именно, в случае неоднородности намагниченности магнита, наличия неоднородностей в материале магнита и т.д. В работах [9–13] и др. использован на практике этот метод. Отметим, что данный метод используют также при настройке (доводке до оптимальной) однородности поля постоянного магнита МР-томографа, когда зарегистрированные в измерениях неоднородности поля устраняют путем внесения дополнительных ферромагнитных вкладывшей.

Метод диполей [25]. В этом методе используются элементарные магниты – *диполи* [2, 18, 25]. Диполь создает в некоторой точке A магнитную напряженность [24, 25]

$$d\mathbf{H} = p \left(\frac{3(\mathbf{e}_p, \mathbf{r})\mathbf{r}}{r^5} - \frac{\mathbf{e}_p}{r^3} \right), \quad (6)$$

где $\mathbf{p} = m d\mathbf{l}$ – магнитный момент диполя, m – воображаемый магнитный заряд, $d\mathbf{l}$ – расстояние между двумя зарядами, образующими диполь, r – расстояние от диполя до точки A , $\mathbf{e}_p$ – единичный вектор вдоль $\mathbf{p}$. Формулу (6) можно записать иначе:

$$d\mathbf{H} = p \frac{3\cos(\mathbf{p}, \mathbf{r})\mathbf{e}_r - \mathbf{e}_p}{r^3}, \quad (7)$$

где $\mathbf{e}_r$ – единичный вектор вдоль $\mathbf{r}$.

Для сравнения приведем формулу для напряженности магнитного поля, создаваемого элементом $d\mathbf{l}$ проводника с током I в некоторой точке A (закон Био–Савара–Лапласа) [18]:

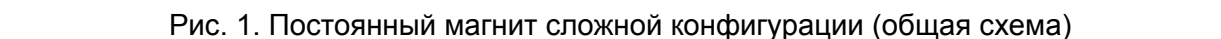
$$d\mathbf{H} = \frac{\mu_0}{4\pi} I \frac{[d\mathbf{l} \times \mathbf{r}]}{r^3}, \quad (8)$$

где $\mu_0 = 4\pi \cdot 10^{-7}$ Тлм/А – магнитная постоянная. Формула (8) часто использовалась для расчета магнитных полей, создаваемых различными проводниками с током [1–6, 18–20]. Формула (6) или (7) может быть также использована для расчета магнитных полей не отдельных диполей, а намагниченных тел (например, постоянных магнитов) путем интегрирования по телу. В частности, если тело (магнит) намагничено однородно, т.е. его намагниченность $\mathbf{J} = \text{const}$, то магнитный потенциал, создаваемый всем телом в некоторой точке A , равен [24, 25] $u = -(\mathbf{J}, \nabla V)$, где V – гравитационный потенциал, создаваемый однородным телом. Видим, что метод диполей является также трудоемким, однако он не требует экспериментальных измерений, как метод потенциала.

Во многих работах ([1–3] и др.) использован также метод, называемый *методом эквивалентного соленоида*. Этот метод основан на аналогии между полем магнита и полем поверхностных некомпенсированных амперовых токов и, как следствие, на возможности рассматривать постоянный магнит как соленоид. Однако этот метод развит лишь для случая, когда намагниченность магнита $\mathbf{J} = \text{const}$, а таким свойством обладают лишь магниты эллипсоидальной формы или магниты с очень высокой коэрцитивной силой и остаточной намагниченностью [1–3]. Кроме того, метод эквивалентного соленоида практически не развит для случая, когда концы магнита имеют углубления сложного профиля. Наконец, метод эквивалентного соленоида в основном использовался для решения прямой задачи – расчета магнитного поля по заданной форме и размерам магнита, а для практики более важна обратная задача – расчет конфигурации магнита, при которой создается наиболее однородное поле. Данная работа посвящена дальнейшей разработке метода [15, 16], который уместно назвать *методом экви-*

Суть метода эквивалентных витков

Рассмотрим *постоянный магнит сложной конфигурации* (рис. 1) – осесимметричный магнит в виде двух половин с зазором и наконечниками, внутри которых должны быть сформированы углубления сложного профиля.



193

отыскании закона $R(z')$ (с ограничениями). При этом полагаем, что намагниченность магнита $\mathbf{J}$ и плотность материала магнита одинаковы по всему объему магнита.

Напомним основные (известные) формулы для напряженности полей витков с током. Заметим, что формулы для полей соленоидов нам не потребуются.

Основные формулы для напряженности магнитного поля витков с током. Рассмотрим *тонкий круговой виток с током* (рис. 2). Продольная составляющая вектора магнитной напряженности, создаваемой витком в некоторой точке $M(z, r)$, равна [2, 3, 5, 6, 15, 19] (в Тл)

$$H_z(z, r) = \frac{C_1}{\sqrt{(R+r)^2 + \Delta z^2}} \left[\frac{R^2 - r^2 - \Delta z^2}{(R-r)^2 + \Delta z^2} E(k) + K(k) \right], \quad (9)$$

где $C_1 = \mu_0 I / 2\pi$, I – ток в витке (в А), R – радиус витка (в м), $\Delta z = z - z'$, причем z' – z -координата центра витка (в м), E и K – полные эллиптические интегралы соответственно 2-го и 1-го рода с модулем

$$k = \left\{ 4Rr / [(R+r)^2 + \Delta z^2] \right\}^{1/2}, \quad (10)$$

равные [27]

$$E(k) = \int_0^{\pi/2} \sqrt{1 - k^2 \sin^2 \psi} \, d\psi, \quad k \in [0, 1], \quad (11)$$

$$K(k) = \int_0^{\pi/2} \frac{d\psi}{\sqrt{1 - k^2 \sin^2 \psi}}, \quad k \in [0, 1), \quad (12)$$

причем $E(0) = K(0) = \pi/2$, $E(1) = 1$, $K(1) = \infty$.

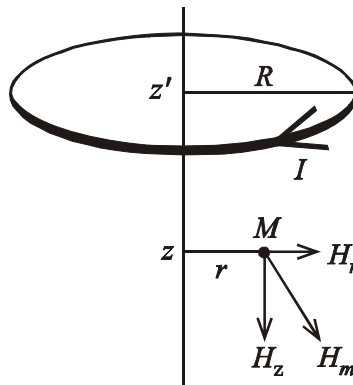


Рис. 2. Тонкий круговой виток с током

Поперечная (радиальная) составляющая равна [5, 19]

$$H_r(z, r) = \frac{C_1 \Delta z}{r \sqrt{(R+r)^2 + \Delta z^2}} \left[\frac{R^2 + r^2 + \Delta z^2}{(R-r)^2 + \Delta z^2} E(k) - K(k) \right]. \quad (13)$$

Часто [6] поперечную составляющую не учитывают, однако в некоторых областях пространства (вблизи витка) она может быть сопоставима с продольной составляющей, поэтому мы ее учитываем.

Модуль вектора магнитной напряженности, который мы обозначим через $H_m(z, r)$, равен

$$H_m(z, r) = |\mathbf{H}(z, r)| = \sqrt{H_z^2(z, r) + H_r^2(z, r)}. \quad (14)$$

Частные случаи. Если точка M находится на оси витка (т.е. $r = 0$) и $R \neq 0$, то $k = 0$ и

$$H_z(z,0) = C_1 \pi \frac{R^2}{(R^2 + \Delta z^2)^{3/2}}, \quad H_r(z,0) = 0, \quad H_m(z,0) = H_z(z,0). \quad (15)$$

Если точка M находится на витке (т.е. $\Delta z = 0$, $r = R$), то

$$H_z(z = z', r = R) = H_r(z = z', r = R) = H_m(z = z', r = R) = \infty. \quad (16)$$

Формулы (9)–(16), справедливые для отдельных круговых витков с током, могут быть использованы для расчета магнитного поля постоянного магнита сложной конфигурации, если представить такой магнит в виде набора (фиктивных) круговых витков с током. При этом нужно учитывать, что магнитное поле витков, уложенных в углубления, вычитается из магнитного поля витков, уложенных на боковые поверхности магнита и наконечников, поскольку направления некомпенсированных амперовых токов у них различные. Аналогично, поле витков, уложенных в «ямки», также вычитается из магнитного поля витков, уложенных на боковые поверхности магнита и наконечников, а поле витков, уложенных на «горки», складывается с магнитным полем витков, уложенных на боковые поверхности магнита и наконечников.

Магнит сложной конфигурации

Рассмотрим магнит в виде двух цилиндров с полюсными наконечниками (pole pieces, pole segments) и зазором (air gap). Для повышения однородности поля в работе [8], а также [9, 10, 12, 13] и др. предложены различные варианты полюсных наконечников: с кольцевым выступом (annular projection) трапецеидальной, треугольной или кривообразной формы вдоль периметра наконечника, с выпуклым выступом (convex projection) дискообразной формы в центре наконечника, а также с выемкой (recess) в середине выпуклого выступа. В этих работах параметры магнита и полюсных наконечников подбирались, а затем магнитное поле рассчитывалось согласно вышеприведенному методу скалярного потенциала.

В данной работе рассмотрим магнит (рис. 1) в виде двух цилиндров с полюсными наконечниками и зазором. Внутри каждого наконечника сформируем углубление (hol-low), а внутри каждого углубления – «ямку» (small pit), или выемку. В работах [9, 10, 13] внутри углубления формируется «горка» (hill), или выпуклый выступ, однако «ямка» (выемка), как показало моделирование, дает более однородное поле. Такой магнит с зазором, углублениями и «ямками» (выемками) или «горками» будем называть *постоянным магнитом сложной конфигурации*.

Что касается ярма (кожуха, оболочки) магнита, то в работах [8, 9] рассмотрено несколько его вариантов: ярмо (yoke) цилиндрической формы с входным отверстием, ярмо с двумя пластинами и несколькими стержнями, ярмо в виде четырехсторонней рамы с входом и др. В работе [8] предложены также устройства для усиления магнитного поля и регулирующий механизм для изменения некоторых параметров магнита (величины зазора, диаметра полюсных наконечников и др.). Однако в данной работе мы рассмотрим только метод определения оптимальной формы магнита, обеспечивающей высокую однородность поля.

Параметры магнита и углублений. Введем следующие обозначения. *Параметры магнита:* L – длина магнита, δ – длина наконечника, R_1 – радиус наконечника, R_2 – радиус магнита, Δ – полудлина зазора. Формы углублений и выемок аппроксимируем прямыми линиями. Прямолинейная аппроксимация наиболее проста при математической и технической реализации, а также порождает наиболее устойчивый алгоритм. Тем не менее, в дальнейшем необходимо рассмотреть и более сложные аппроксимации.

Прямолинейной аппроксимации соответствует следующая формула для радиуса витка, уложенного в углубление, в функции z' (см. рис. 3):

$$R(z') = \frac{\sigma}{g}(\Delta - |z'|) + \rho + \sigma, \quad (17)$$

и формула для радиуса витка, уложенного в выемку:

$$R(z') = \frac{\xi}{p}(\Delta + g - |z'|) + \eta + \xi. \quad (18)$$

Параметры углубления: глубина g , минимальный радиус ρ , максимальный радиус $\rho + \sigma$; *параметры выемки:* глубина p , минимальный радиус η , максимальный радиус $\eta + \xi$.

Параметры магнита L , δ , R_1 , R_2 , Δ будем считать заданными, а параметры углубления g , ρ , σ , p , η , ξ подлежащими определению.

Критерий выбора параметров углублений. В качестве *критерия выбора* оптимальных значений g , ρ , σ , p , η , ξ используем условие минимума отклонения рассчитанного поля от однородного [15, 16]:

$$\varepsilon_{\text{opt}} = \min_{g, \rho, \sigma, p, \eta, \xi} \varepsilon, \quad (19)$$

где

$$\varepsilon = \frac{1}{n+1} \sum_{i=0}^n \left| \frac{H_z(z_i, 0) - H_z(0, 0)}{H_z(0, 0)} \right|, \quad (20)$$

причем n – число дискретных шагов h вдоль z от $z = 0$ до $z = \Delta/2$. Задачу минимизации (19) можно решать, используя известные методы минимизации функционалов без ограничений и с ограничениями на искомые параметры (градиента, Ньютона, координатного спуска, проекции градиента и др.) [28].

При этом *ограничения* на параметры целесообразно задавать в виде:

$$(q_i)_{\min} \leq q_i \leq (q_i)_{\max}, \quad i = 1, \dots, 6, \quad (21)$$

где $q_1 = g$, $q_2 = \rho$, $q_3 = \sigma$, $q_4 = p$, $q_5 = \eta$, $q_6 = \xi$ – искомые параметры, а значения $(q_i)_{\min}$, $(q_i)_{\max}$ следует выбирать, исходя из физико-технических ограничений на параметры, из решения близких задач и т.д.

Результаты численного моделирования

Разработан *пакет программ* MAGNET для определения оптимальных конфигураций цилиндрических магнитов с зазорами и наконечниками, имеющими углубления и выемки (с целью повышения однородности полей), и для расчета их магнитных полей методом эквивалентных витков. Программы разработаны на MS Fortran 5 и Fortran 90, а графика – на MathCAD, CorelDRAW и PaintBrush. Вычисления запрограммированы с двойной точностью.

Рассмотрим *пример магнита* (рис. 3), имеющего следующие *параметры* (в мм): $L = \delta = 100$, $R_1 = R_2 = 320$, $\Delta = 150$. Это – заданные параметры.

Для расчета компонент поля $H_z(z, r)$, $H_r(z, r)$ и $H_m(z, r)$ в зазоре задавалась *сетка узлов*: $z \in [0, z_{\max}]$, $r \in [0, r_{\max}]$ с шагом дискретизации $h = \Delta z = \Delta r = 1$ мм, где $z_{\max} = r_{\max} = 140$ мм. Суммирование по (фиктивным) виткам магнита, наконечников, углублений и выемок производилось с шагом дискретизации $\Delta z' = 0.5$ мм.

Сначала рассчитывалась функция $H_z(z, 0)$ согласно (15) путем суммирования по (фиктивным) виткам, причем поля витков углублений и выемок вычитались из полей витков боковых поверхностей магнита и его наконечников. Данный расчет не требует больших затрат компьютерного времени в виду простоты формулы (15).

Были получены следующие *оптимальные значения параметров* (в мм):
 $g = g_{\text{opt}} = 81.0$, $\rho = \rho_{\text{opt}} = 209.7$, $\sigma = \sigma_{\text{opt}} = 90.6$, $p = p_{\text{opt}} = 3.5$, $\eta = \eta_{\text{opt}} = 85.5$,
 $\xi = \xi_{\text{opt}} = 1.4$, $\varepsilon = \varepsilon_{\text{opt}} = 2.719 \cdot 10^{-6} \approx 2.72$ ppm.

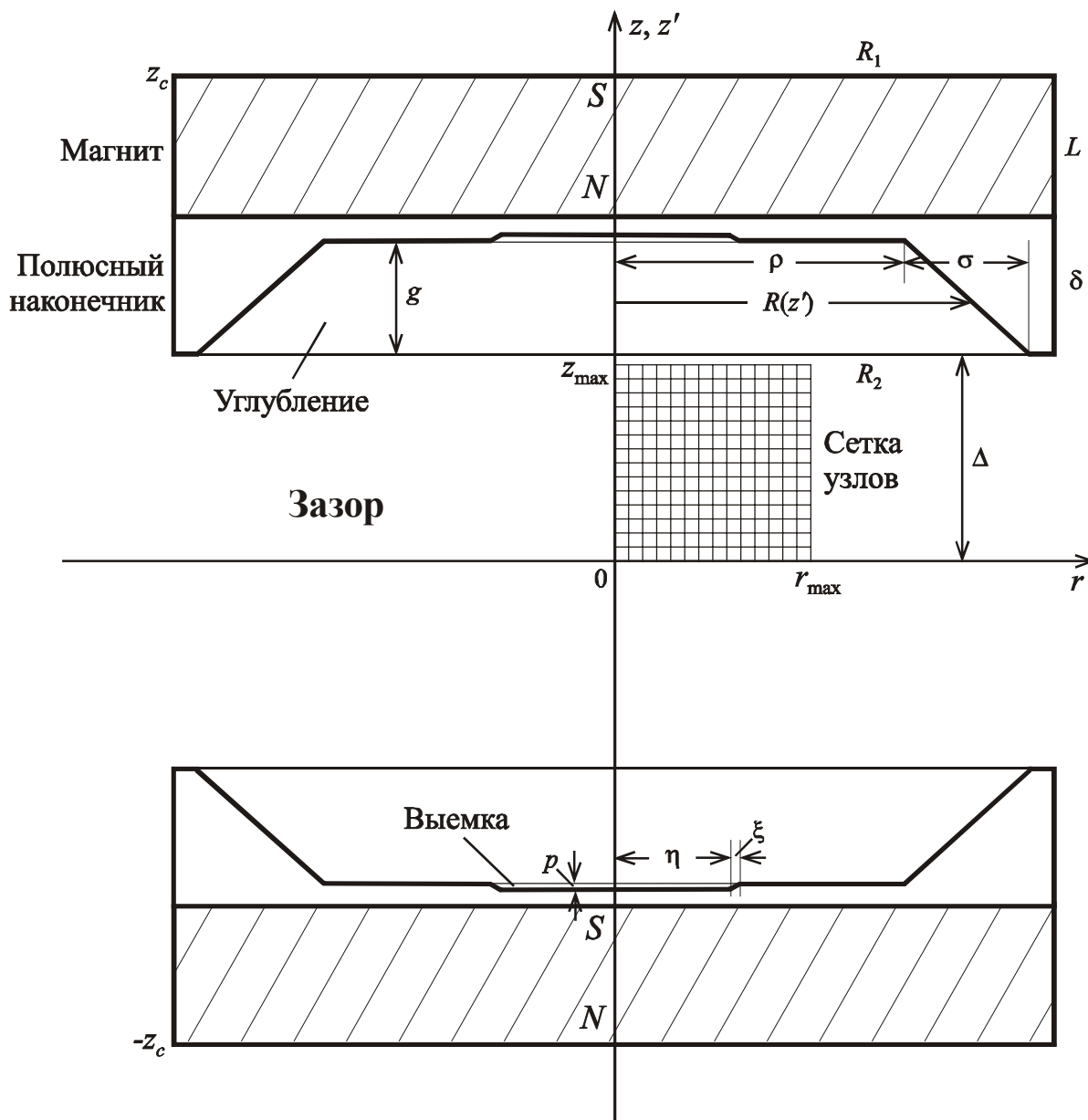


Рис. 3. Пример магнита

Затем при найденных оптимальных значениях параметров была рассчитана функция $H_z(0, r)$ согласно (9). На рис. 4 приведены две нормированные функции. $H_z(z, 0)/H_z(0, 0)$ и $H_z(0, r)/H_z(0, 0)$ при оптимальных значениях параметров. Результаты, отображенные на рис. 4, качественно близки результатам японских физиков [9, 10, 12, 13]. Однако в работах [9, 10, 12, 13] в основания углублений были введены так называемые «горки» (выпуклые выступы), а в данной работе введены «ямки» (выемки), что позволило получить более однородное поле.

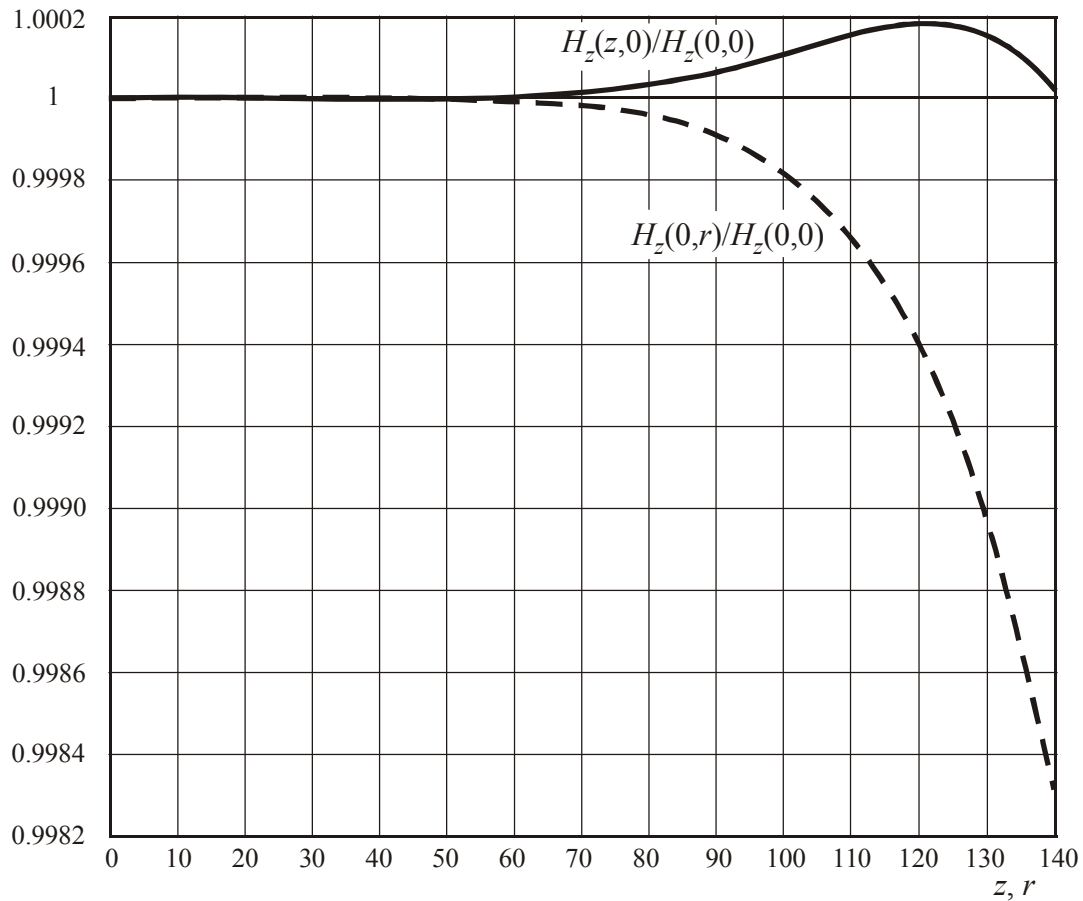


Рис. 4. Нормированные функции $H_z(z,0)/H_z(0,0)$ и $H_z(0,r)/H_z(0,0)$

Далее были рассчитаны компоненты поля $H_z(z,r)$, $H_r(z,r)$ и $H_m(z,r)$ на всей сетке узлов z, r согласно формулам (9)–(14) – этот расчет требует наибольших затрат компьютерного времени. На рис. 5 приведены изолинии нормированного поля $H_m(z,r)/H_m(0,0)$. Эти результаты качественно близки результатам работы [13].

На рис. 6 для большей наглядности и информативности приведены изолинии функции (логарифм относительной неоднородности поля)

$$\lg \frac{|H_m(z,r) - H_m(0,0)|}{H_m(0,0)}, \quad (22)$$

причем непрерывными линиями отображены изолинии, а пунктирной линией – 50-процентная зона (рабочая зона радиуса $\Delta/2$). Видим, что рис. 6 дает более точное представление о неоднородности поля, чем рис. 5 или 4.

Замечания

Приведем следующие важные замечания математического и технического характера.

1. Если значения всех параметров $L, \delta, R_1, R_2, \Delta, g, \rho, \sigma, p, \eta, \xi$ умножить на некоторый множитель $a > 0$, то, как видно из (9), (13), (15), ход относительных зависимостей $H_z(z,r)/H_z(0,0)$, $H_r(z,r)/H_r(0,0)$, а значит, и вид кривых на рис. 4–6 не изменится. Лишь z и r нужно умножить на a . Другими словами, если решен некоторый частный численный пример, то из него можно получить ряд других примеров путем умножения всех параметров на ряд значений a (без решения этих примеров). Например, положив $a = 1.4$, получим новые значения параметров (в мм): $L = \delta = 140$,

$R_1 = R_2 = 448$, $\Delta = 210$, $g = 113.4$, $\rho = 293.58$, $\sigma = 126.84$, $p = 4.9$, $\eta = 119.7$, $\xi = 1.96$, $h = 1.4$, $z_{\max} = r_{\max} = 196$, $\Delta z' = 0.7$, $\Delta/2 = 105$. При этом ε сохранит свое значение: $\varepsilon = 2.719 \cdot 10^{-6}$, а также сохраняют свой вид кривые на рис. 4–6. При этом сложных пересчетов делать не нужно. Данное замечание согласуется с *теоремами подобия* магнитных систем [3, с. 108].

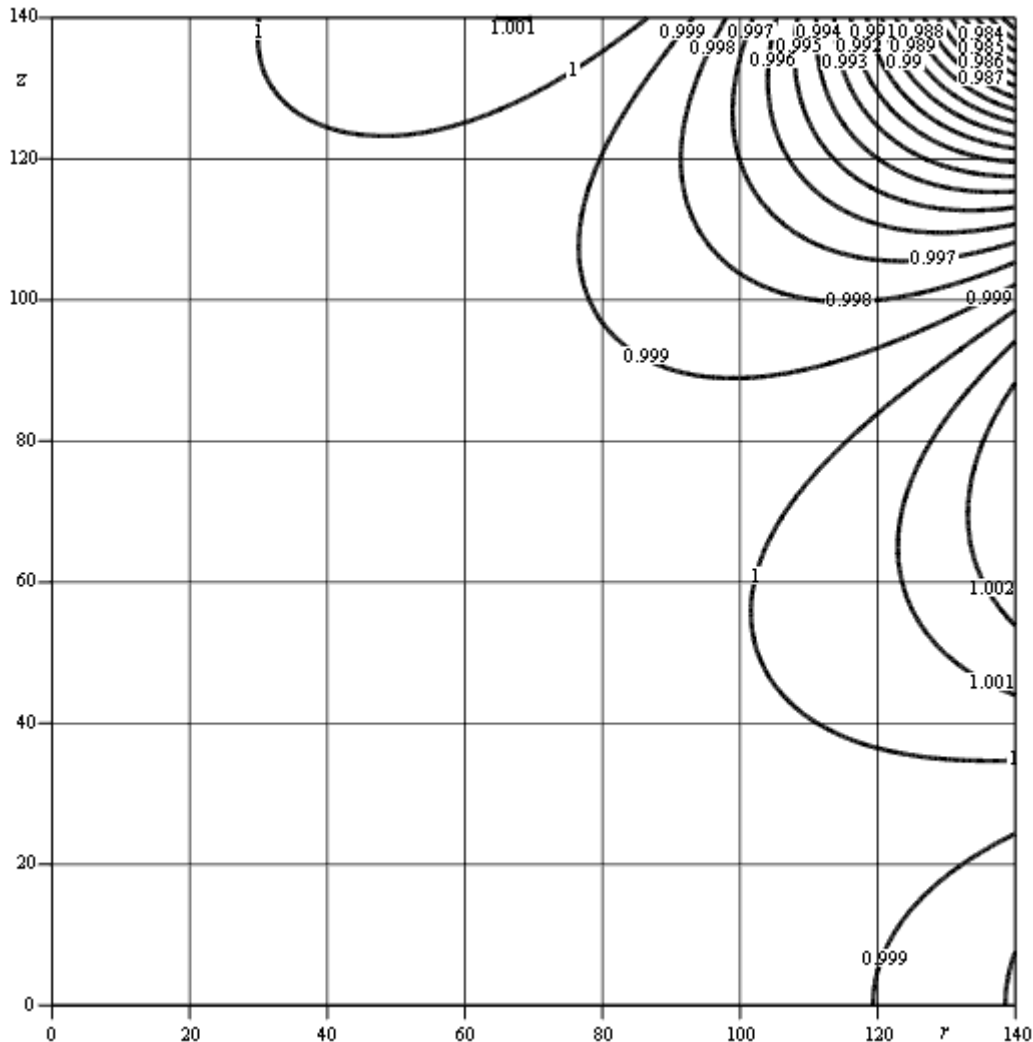


Рис. 5. Изолинии нормированного поля $H_m(z, r)/H_m(0, 0)$

2. Если постоянный магнит изготовить из мягкого железа, то это будет дешевый магнит, но мягкое железо быстро размагничивается, и потребуются непрерывное намагничивание его, например, с помощью катушки с током простой конфигурации. Кроме того, у магнита из мягкого железа будет иметь место неоднородность намагниченности J . Если же использовать высококачественный сплав, например, Nd–Fe–B (как в работах [8–10, 12, 13]), Sm–Co, РЗМ, Альнико и т.д., то это будет дорогой магнит, но для него рассмотренный метод эквивалентных витков будет давать достаточно точные результаты (поскольку в этом случае практически $\mathbf{J} = \text{const}$).

3. Метод эквивалентных витков изложен применительно к идеализированному случаю однородного намагничивания магнита ($\mathbf{J} = \text{const}$), что справедливо лишь для магнита эллипсоидальной формы или магнита из материала с очень большой коэрцитивной силой и остаточной намагниченностью [3, с. 203]. Кроме того, полагается, что магнит и его наконечники являются однородными. Однако на практике часто имеют

место заметные неоднородности материала магнита и наконечников. В этих случаях расчеты параметров магнита и поля данным методом можно уточнить методом скалярного потенциала или иным методом (способом), использующим измерения поля магнитометром. При этом расчеты методом эквивалентных витков можно использовать в качестве хорошего начального приближения.

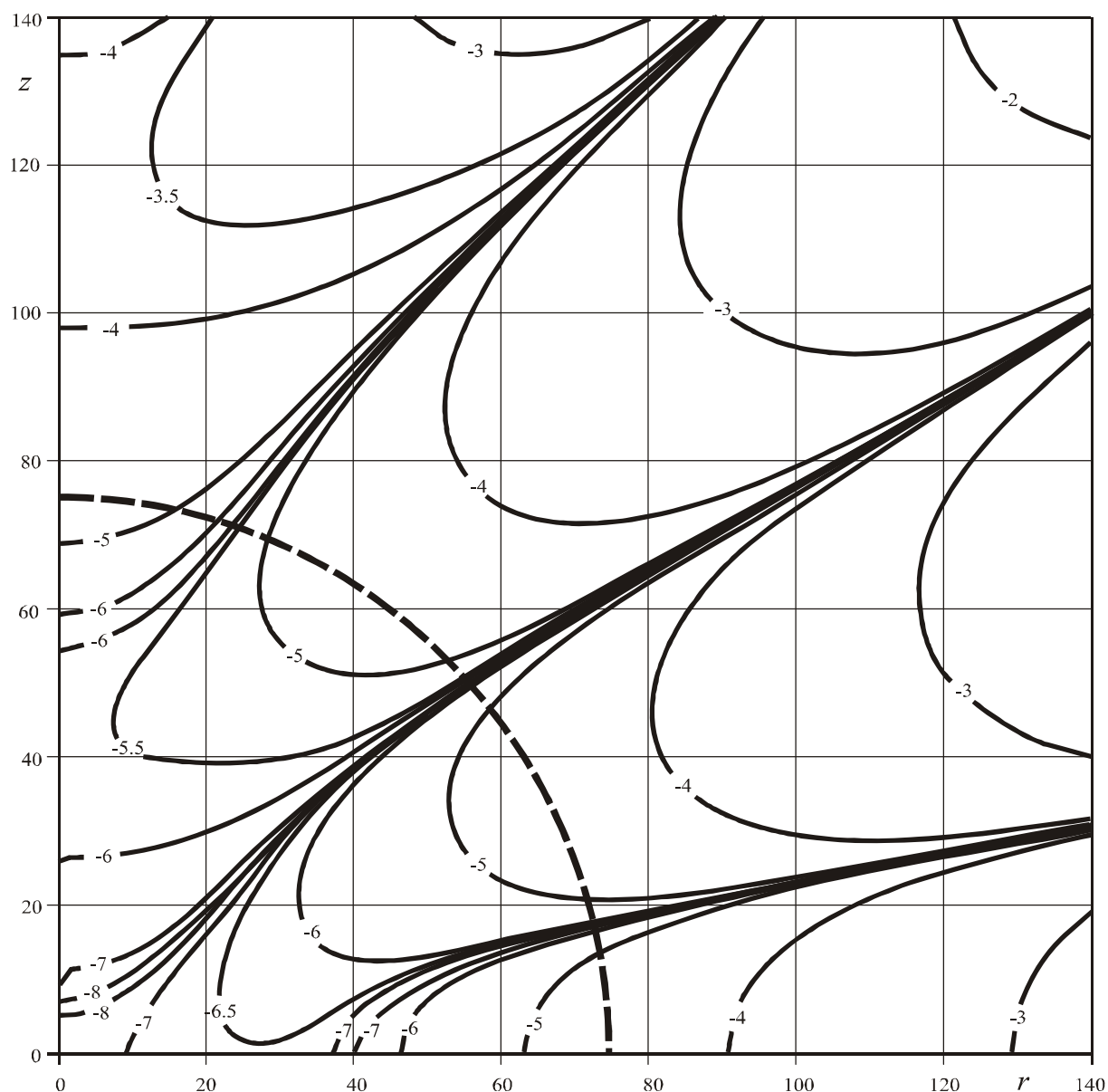


Рис. 6. Изолинии логарифма относительной неоднородности поля

4. В полюсных наконечниках имеют место токи Фуко [18, с. 430], которые уменьшают добротность резонансной индуктивности, принимающей МР-сигналы. Для подавления токов Фуко предлагается изготавливать полюсные наконечники из набора изолированных стержней, параллельных оси магнита. При этом стержни целесообразно изготавливать прямоугольными в сечении, чтобы они плотно прилегали друг к другу.

Заключение

В данной работе дано развитие метода эквивалентных витков расчета конфигураций постоянных магнитов МР-томографов, создающих высокооднородные поля. Для достижения высокой однородности поля в наконечниках магнитов использованы углубления и выемки («ямки»). Для расчета полей использована *аналогия* между посто-

янным магнитом (а также его наконечниками, углублениями и выемками) и набором витков с током. Приведены рабочие формулы, разработаны программы и решены модельные примеры, показавшие, что данный метод позволяет в принципе (при условии постоянства намагниченности и плотности материала магнита) получать высокооднородные поля постоянных магнитов с относительной неоднородностью $\Delta H/H \sim 10^{-5} - 10^{-6}$, т.е. 1–10 ppm в рабочей области, причем необходимая конфигурация магнита определяется путем чисто компьютерного моделирования без измерений поля.

Сопоставим результаты данной работы с результатами работы [15]. В работе [15] положено $L = \delta = 125$, $R_1 = R_2 = 300$, $\Delta = 150$ и получено $\varepsilon = 4.493 \cdot 10^{-6} \approx 4.5$ ppm, т.е. в данной работе получено более однородное поле, чем в работе [15]. Если проанализировать также результаты работ [9, 10, 12, 13, 16], то можно сделать вывод, что однородность поля повышается ($\Delta H/H$ понижается) с уменьшением отношения Δ/R , где $R = (R_1 + R_2)/2$, а также с уменьшением $L + \delta$. Можно рекомендовать брать отношение Δ/R примерно в диапазоне 0.43–0.47.

Отметим также, что в работах [9, 13] достигнута (методом скалярного потенциала) однородность поля ≈ 30 –70 ppm, а в наших работах (в [15, 16] и в данной работе) ≈ 3 –5 ppm, т.е. получено более однородное магнитное поле, причем примерно в одинаковых рабочих областях.

Метод эквивалентных витков можно рекомендовать для практического использования при быстром компьютерном расчете постоянного магнита сложной конфигурации применительно к МР-томографу.

Работа выполнена при финансовой поддержке РФФИ (грант № 05-08-01304-а).

Литература

1. Коген-Далин В.В., Комаров Е.В. Расчет и испытание систем с постоянными магнитами. М.: Энергия, 1977. 248 с.
2. Афанасьев Ю.В., Студенцов Н.В., Хорев В.Н. и др. Средства измерений параметров магнитного поля. Л.: Энергия, 1979. 320 с.
3. Постоянные магниты. Справочник. Изд-е 2-е / Под ред Ю.М. Пятина. М.: Энергия, 1980. 488 с.
4. Lugansky L.B. Optimal coils for producing uniform magnetic fields // J. Phys. E: Sci. Instrum. 1987. Vol. 20. P. 277–285.
5. Тихонов А.Н., Рубашов И.Б., Арсенин В.Я. и др. О математическом проектировании конструкции ЯМР-томографа. Препринт ИПМ АН СССР, 1987. 24 с.
6. Галайдин П.А., Иванов В.А., Марусина М.Я. Расчет и проектирование электромагнитных систем магниторезонансных томографов. СПб.: СПбГУ ИТМО, 2004. 87 с.
7. Сизиков В.С. Математические методы обработки результатов измерений. СПб.: Политехника, 2001. 240 с.
8. Miyamoto T., Sakurai H., Hayashi H., Ohnishi Y. Magnetic field generating device for NMR-CT // Patent US4672346. Cl. B29C67/20, C08J5/18, G01R33/383. 1987.
9. Miyamoto T., Sakurai H., Takabayashi H., Aoki M. Development of a permanent magnet assembly for MRI // J. Magnet. Soc. Japan. 1989. Vol. 13. N 2. P. 465–468.
10. Miyamoto T., Sakurai H., Yanaka S. et al. Application of Nd-Fe-B magnet to MRI systems // J. Magnet. Soc. Japan. 1989. Vol. 13. N 4. P. 567–571.
11. Abele M.G., Chandra R., Rusinek H. et al. Compensation of non-uniform magnetic properties of components of a yokeless permanent magnet // IEEE Trans. on Magnetism. 1989. Vol. 25. N 5. P. 3904–3906.

12. Miyamoto T., Sakurai H., Takabayashi H., Aoki M. A development of a permanent magnet assembly for MRI devices using Nd-Fe-B material // IEEE Trans. on Magnetics. 1989. Vol. 25. N 5. P. 3907–3909.
13. Sakurai H., Aoki M., Miyamoto T. Improvement of the field homogeneity with a permanent magnet assembly for MRI // J. Magnet. Soc. Japan. 1990. Vol. 14. N 2. P. 465–468.
14. Марусина М.Я., Казначеева А.О. Современные виды томографии. СПб.: СПбГУ ИТМО, 2006. 152 с.
15. Сизиков В.С., Соколов Д.Ю. О синтезе высокооднородного поля постоянного магнита в МР-томографии // Научное приборостроение. 2006. Т. 16. № 4. С. 97–104.
16. Сизиков В.С., Соколов Д.Ю. О повышении однородности поля постоянного магнита МР-томографа // Изв. вузов. Приборостроение. 2006. Т. 49. № 12. С. 32–38.
17. Неронов Ю.И., Каршенбойм С.Г. Метод определения частоты резонанса ЯМР-сигналов и регистрация эффектов изотопного замещения ядер молекул водорода // Научное приборостроение. 2006. Т. 16, № 1. С. 97–101.
18. Фриш С.Э., Тиморева А.В. Курс общей физики. Учебник. Т. 2. Электрические и электромагнитные явления. Изд-е 10-е. СПб.: Лань, 2006. 528 с.
19. Дружкин Л.А. Задачи теории поля. М.: МИРГЭ, 1964. 462 с.
20. Монтгомери Д. Получение сильных магнитных полей с помощью соленоидов. М.: Мир, 1971. 360 с.
21. Тамм И.Е. Основы теории электричества. Уч. пособие для вузов. Изд-е 11-е. М.: ФИЗМАТЛИТ, 2003. 616 с.
22. Поливанов К.М. Теоретические основы электротехники. Ч. 3. Теория электромагнитного поля. М.: Энергия, 1969. 352 с.
23. Фейнман Р., Лейтон Р., Сэндс М. Фейнмановские лекции по физике. Т. 5. Электричество и магнетизм. Изд-е 2-е. М.: Мир, 1977. 302 с.
24. Парселл Э. Электричество и магнетизм. Берклевский курс физики. Учебник для вузов. Изд-е 4-е. СПб.: Лань, 2005. 416 с.
25. Яновский Б.М. Земной магнетизм. Изд-е 4-е. Л.: ЛГУ, 1978. 592 с.
26. Корзюк В.И., Шушкевич Г.Ч. Проникновение магнитного поля в тонкую незамкнутую эллипсоидальную оболочку в присутствии проводящей толстостенной сферической оболочки // ЖТФ. 2006. Т. 76. Вып. 6. С. 9–14.
27. Бронштейн И.Н., Семендяев К.А. Справочник по математике для инженеров и учащихся втузов. Изд-е 13-е. М.: Наука, 1986. 544 с.
28. Химмельблау Д. Прикладное нелинейное программирование. М.: Мир, 1975. 536 с.

**РЕКОНСТРУКЦИЯ ТРЕХМЕРНЫХ МОДЕЛЕЙ
ПО РАСТРОВЫМ ИЗОБРАЖЕНИЯМ****А.В. Меженин, В.Т. Тозик**

Рассматривается проблема реконструкции трехмерных моделей для случаев использования как калиброванных, так и некалиброванных изображений. Для выделения контуров на изображениях предложено использовать метод Собеля и фильтры Габора. Проведенные эксперименты подтвердили эффективность рассматриваемых методов.

Введение

Нахождение реальных размеров объектов (сложных корпусов, профилей, натуральных моделей и т.д.) для получения трехмерных виртуальных моделей является актуальной задачей. Один из подходов – использование специальных устройств для трехмерного бесконтактного сканирования. Однако их стоимость в настоящее время непомерно высока. Кроме этого, не всегда исследуемые объекты можно поместить в такие устройства. Другой подход – использование растровых изображений исследуемого объекта и реконструкция на их основе трехмерной модели.

Этот метод относится к науке об определении форм, размеров и положения объектов по их фотографическим изображениям – *фотограмметрии*. В настоящее время техническое и программное обеспечение фотограмметрии применяется в геодезии, картографии, военном деле, космических исследованиях, для создания 3D-моделей промышленных установок, в архитектуре и др.

В обычном изображении трехмерного пространства информация о расстоянии до различных элементов сцены проявляется только в виде косвенных признаков – через относительные размеры объектов, затенение одних объектов другими, различную освещенность и т.д. Один из способов получения информации о глубине состоит в регистрации нескольких изображений сцены под различными ракурсами. В этом случае точки сцены дают изображения, относительное положение которых зависит от расстояния до точки наблюдения. Сопоставляя эти изображения, в ряде случаев можно реконструировать трехмерную структуру сцены.

Таким образом, основной задачей проводимых исследований является разработка методов и программных средств для реконструкции трехмерных моделей на основе растровых изображений исследуемого объекта. Кроме этого, важной задачей является оценка эффективности различных алгоритмов распознавания. Для получения объективных данных об эффективности того или иного алгоритма необходимо иметь не только наборы калиброванных изображений, но и рассматривать различные условия получения исходных растровых изображений.

Существующие реализации

Техническое и программное обеспечение для промышленной фотограмметрии предлагают различные известные фирмы, такие как Intergraph или Siemens. Однако одну из самых недорогих и простых технологий предлагает американская фирма Vexcel. Программное обеспечение фирмы Vexcel, работая прямо в среде AutoCAD или MicroStation совместно с любыми подходящими специализированными программами, позволяет по фотограмметрической базе в полуавтоматическом режиме воссоздать

трехмерную модель установки, а по ней – и все необходимые чертежи. Полученную модель вместе с фотограмметрической базой можно использовать для последующей реконструкции, а также создания различных информационных систем для действующих производств.

Большие работы по обработке изображений и реконструкция моделей зданий ведутся на кафедрах и в лабораториях ведущих университетов мира. Это Cambridge University под руководством профессора Roberto Cipolla [1], Image Processing Laboratory Computer Science Department – University of Catania под руководством Roberto Scopigno [2] и Berkley University под руководством Paul E. Debevec.

Этапы реконструкции

Выделим основные этапы реконструкции.

1. Выявление характерных точек на изображении
 - 1.1. Получение нормализованного полутонового изображения.
 - 1.2. Поиск исследуемых областей.
 - 1.3. Выделение краев на исследуемой области (методы Собеля, Лапласа, Канни).
 - 1.4. Преобразование рассматриваемого участка в монохромное изображение.
 - 1.5. Анализ полученного монохромного и полутонового изображения в исследуемой области.
2. Сопоставление точек полученных на различных растровых изображениях
3. Получение трехмерной модели и анализ полученных размеров

Вычисление трехмерных координат точки по двум изображениям

Рассмотрим ситуацию, когда две камеры, находящиеся в разных точках, регистрируют одну и ту же сцену. Пара изображений, получаемых при этом, называется стереопарой. Рассмотрим простейший случай (рис. 1).

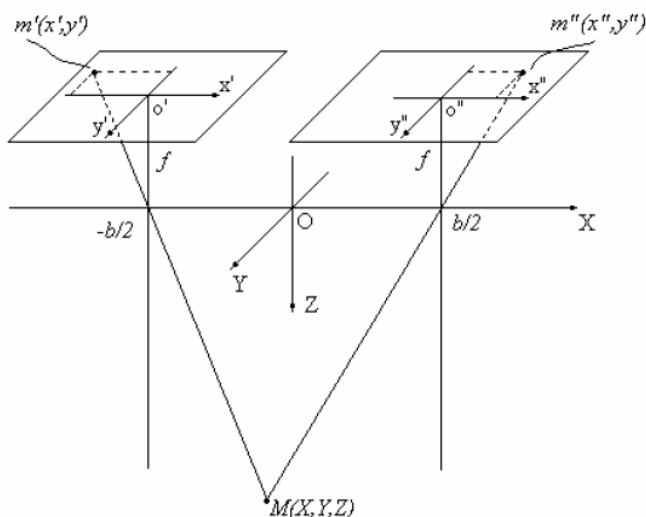


Рис. 1. Простейшая стереоскопическая система

Пусть одинаковые камеры расположены так, что их оптические оси параллельны, а прямая, проходящая через оптические центры, перпендикулярна оптическим осям (эта прямая называется базовой линией, а ее отрезок, заключенный между оптическими центрами – базой). Положим длину базы равной b . Выберем такую глобальную систему координат, начало которой O расположено на базовой линии посередине между опти-

ческими центрами камер, ось OZ параллельна оптическим осям, а ось OX направлена вдоль базовой линии. Пусть начала координат в плоскостях изображений камер совпадают с главными точками ($u_0 = v_0 = 0$), а единицы измерения координат в глобальной системе и в плоскостях изображения камер одинаковы ($w = h = 1$).

Выберем точку М с глобальными координатами (X, Y, Z) . Координаты ее проекции в плоскости изображения первой (левой) камеры обозначим через (x', y') , а в плоскости изображения второй (правой) камеры – через (x'', y'') . (Проекции одной и той же точки М в плоскостях изображений разных камер называются *сопряженными точками*.) Нетрудно проверить, что

$$x' = f(X + b/2)/Z, \quad x'' = f(X - b/2)/Z, \quad y' = y'' = fY/Z.$$

Заметим, что в направлении, перпендикулярном направлению базовой линии, координаты сопряженных точек (у-координаты) совпадают. Это обстоятельство имеет большое значение при автоматизированном поиске сопряженных точек на стереопаре, позволяя существенно сократить размеры зоны поиска. Из первых двух соотношений следует, что

$$Z = fb/(x' - x'').$$

Это означает, что, зная геометрию съемки и выполнив измерения координат проекций одной и той же точки в плоскостях изображения камер, можно вычислить глубину (координату Z) этой точки. Более того, полученные соотношения позволяют вычислить полностью трехмерные координаты точки:

$$X = b \frac{(x' + x'')}{2(x' - x'')}, \quad Y = b \frac{(y' + y'')}{2(x' - x'')}.$$

Из всего изложенного следует, что для оценивания трехмерных координат некоторой точки по стереопаре необходимо:

- а) знать внутренние параметры камер (задача калибровки),
- б) знать параметры взаимного расположения камер (задача взаимного ориентирования),
- в) найти и определить на изображениях координаты соответствующих данной точке сопряженных точек (задача поиска сопряженных точек).

Использование некалиброванных изображений

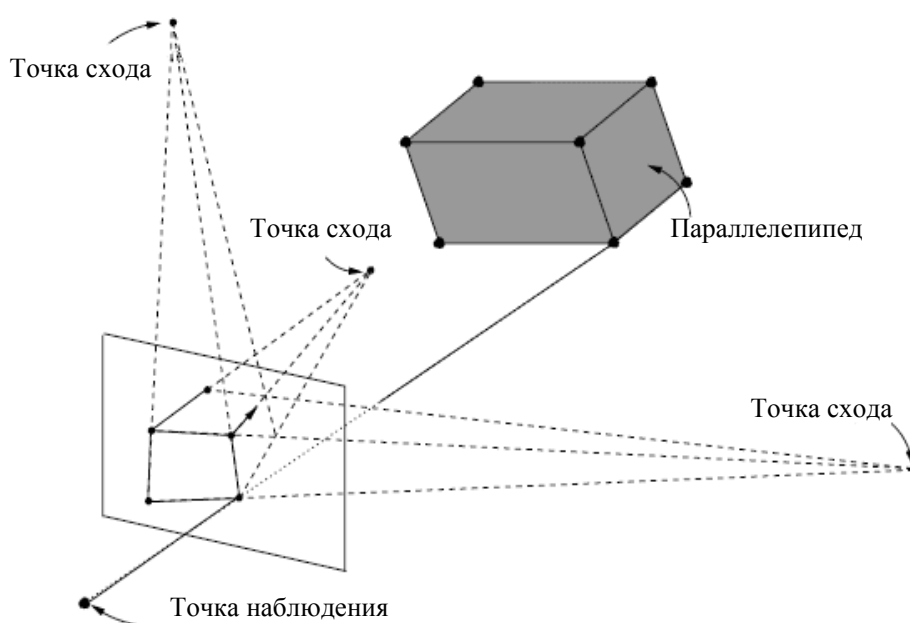


Рис. 2. Реконструкция точек схода

Одной из проблем в решении задачи реконструкции является получение полностью калиброванных изображений, когда для каждого изображения должны быть известны все параметры камеры, с помощью которой оно было получено, включая ее положение и ориентацию относительно объекта. Калибровка изображений – это отдельная сложная задача, которой посвящено множество работ. Иногда ее решение требует использования дополнительных устройств – управляемых вращающихся столов, специальных меток и пр.

Для определенного класса объектов (например, архитектурных) можно воспользоваться следующим наблюдением: многие элементы сцены могут быть аппроксимированы параллелепипедами. В этом случае можно реконструировать точки схода линий, являющихся продолжением ребер объекта (рис. 2). Координаты найденных точек могут служить основой для калибровки изображений, полученных с точек наблюдения, координаты которых неизвестны.

Программная реализация

Рассмотренный математический аппарат был реализован в виде программы Recognition. На рис. 3 представлен этап нахождения габаритных точек объекта.

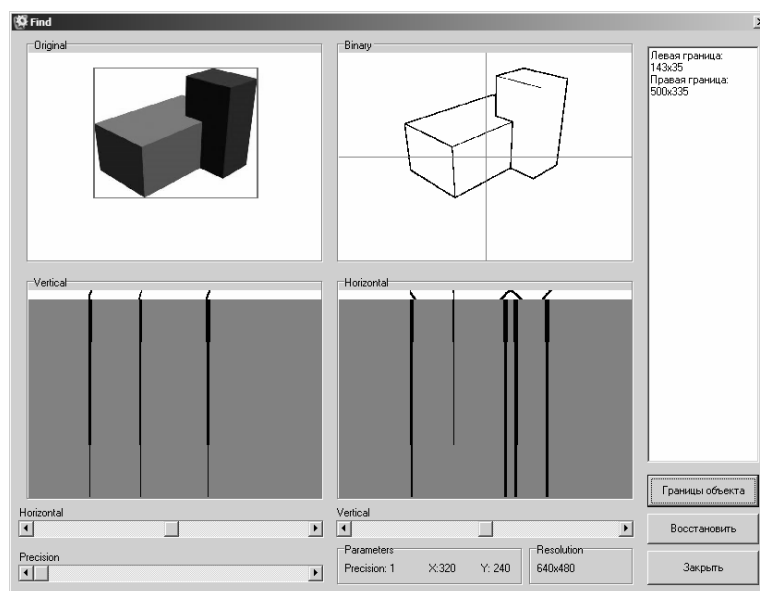


Рис. 3. Нахождение габаритных точек объекта

В качестве исходных изображений были выбраны объекты без помех и варианты с изменением фона и в условиях неравномерного освещения. Следует отметить, что отклонение определения границ данного объекта составляет $\Delta_b = \pm 2$ пикселя, в зависимости от фона и наложения цвета. Поиск границ осуществляется на монохромном изображении. Все точки, кроме точек контура, имеют яркость $G_1 = 255$, а точки контура имеют яркость $G_2 = 0$. Матрица изображения I'' содержит элементы двух типов, поэтому определение границ рассматриваемого объекта не представляет трудности. Осуществляется перебор элементов матрицы изображения, номер первого и последнего элемента (с яркостью $G_2 = 0$) как по вертикали, так и по горизонтали запоминается. Полученные числа и есть границы рассматриваемого объекта.

На рис. 4 представлен результат реконструкции холла здания университета СПбГУ ИТМО.



Рис. 4. Трехмерная модель холла

Заключение

Реализация программного обеспечения и проведенные исследования показали эффективность предложенного подхода. Предложенный метод получения калиброванных изображений позволил протестировать алгоритмы при различных исходных изображениях. Визуальная оценка и измерения полученных координат габаритных размеров показывают эффективность рассмотренных алгоритмов поиска характерных точек на изображении.

Литература

1. <<http://mi.eng.cam.ac.uk/~cipolla/>>
2. <<http://vcg.isti.cnr.it>>
3. 3ds Max 8: трехмерное моделирование и анимация / В.Т. Тозик, А.В. Меженин. – СПб.: БХВ-Петербург, 2006. 1008 с.: ил.
4. Меженин А.В., Тозик В.Т. Современная технология для представления трехмерных моделей в Интернет / Труды X Всероссийской научно-методической конференции «Телематика '2003». СПб, 2003.
5. Popov S.A., Kirichuk V.S. Algorithm of Estimation of the Geometric Parameters of the System of Two Projection Cameras by the Method of the Least Squares (MLS). //Pattern Recognition and Image Analysis. 1999. №2. P.304.
6. Марр Д. Зрение: информационный подход к изучению представления и обработки зрительных образов. М.: Радио и связь, 1987.

ПОЛУЧЕНИЕ КВАЗИСТЕРЕОСКОПИЧЕСКИХ ИЗОБРАЖЕНИЙ

Д.М. Поздеев, А.В. Меженин, В.Т. Тозик

В статье обсуждаются области применения квазистереоскопических изображений и методы их получения. Рассматривается возможность получения объемного изображения на основе использования бинокулярных свойств зрения.

Введение

Целью проводимых исследований является демонстрация возможностей использования стереоскопических и квазистереоскопических изображений в областях науки, использующих пространственные построения:

1. построение геометрических объектов и сложных сечений в стереометрии;
2. моделирование взаимного расположения атомов в молекулах в химии или в кристаллической решетке в структурной кристаллографии;
3. картография;
4. проектирование сложных объемных конструкций.

Традиционные системы САПР позволяют создать цифровой макет изделия, осмотреть его, проанализировать на предмет ошибок и взаимоисключений. Следующая задача – посмотреть, как продукт поведет себя в реальной среде и окружающей обстановке, как он будет взаимодействовать с другими изделиями или как несколько человек одновременно могут пользоваться этим изделием. Для этой цели существует особый класс задач – системы макетирования и виртуальной реальности (Digital Mockup), которые позволяют построить зрительный образ изделия и окружающей обстановки, приближенной к реальному [1]. Подобные технологии применяются в ведущих компаниях как автомобильной, так и авиационной промышленности – BMW, DaimlerChrysler, Airbus, Boeing и т.д.

Методы получения стереоскопических изображений

Следует различать технологии получения стереоскопических и квазистереоскопических изображений. В первом случае существует либо трехмерная модель объекта, либо два изображения этого объекта, полученные с разных точек. Во втором случае (квазистерео) эффект объемности достигается посредством преобразования 2D изображения (видеоконтента).

Основные методы получения стереоскопических изображений – анаглифный (Роллман, 1852 г.), поляризационного и растрового стереовидения, интегральной фотографии, «мозаичных картинок», «Virtual Reality», голографии.

Гибкость электронных средств отображения, к которым относятся телевизор и компьютер, позволяет так обработать сигнал обычного (моноскопического) видеоизображения, чтобы получить новое изображение, максимально приближенное к стереоскопическому по его восприятию зрителем. Наиболее ценно при этом получить сочетание трех важных качеств в используемом средстве отображения – достаточно достоверное воспроизведение исходной объемной сцены в реальном времени (без существенных временных задержек в устройстве обработки) при максимальной простоте. Для получения квазистереоскопического видеоизображения предложено достаточно много методов [2–4]. Наиболее интересны методы, в которых собственно информационный сигнал изображения не затрагивается устройством обработки, а изменяется только геометрическая структура изображения за счет обработки служебных сигналов, находящихся в составе видеосигнала (например, строчных синхросигналов [4]).

Эти методы основаны на особенности восприятия человеком глубины пространства. Вспомним, как художник добивается ощущения объема при рассматривании зри-

телем картин, воспроизводящих на плоском полотне натуральные сцены (природные ландшафты, различные интерьеры). Для натуральных сцен, как правило, характерным является наиболее близкое расположение к зрителю тех объектов, которые находятся внизу сцены (снизу картины), например, земля, а наиболее далеки от зрителя объекты вверху сцены (сверху картины), например, небо. Поэтому художник имитирует разную удаленность предметов относительно зрителя, плавно меняя масштаб изображения по вертикали, т.е. искусственно вносит перспективные искажения (в соответствии с законами перспективы). Естественные перспективные искажения изначально присутствуют в любом моноскопическом изображении, представляющем натуральные сцены (снятые реальной камерой или синтезированные методом трехмерной графики на компьютере). Наблюдаемое любое моноскопическое изображение какой-либо трехмерной сцены, реализованное в соответствии с законами перспективы, человек в силу соответствующих природных особенностей своего зрения всегда ощущает разную удаленность предметов изображаемой сцены.

Следует отметить, что телевизионному (видео) изображению присуща черта, которая отсутствует в картине, нарисованной художником, а именно – динамизм, движение объектов наблюдаемой сцены. А зрению человека присуще свойство, которое можно назвать динамическим эффектом глубины. Оно дополнительно помогает человеку оценивать расстояния до наблюдаемых предметов (чем ближе одни и те же предметы, тем быстрее они движутся). Наконец, человеку присуще бинокулярное зрение. В принципе для оценивания расстояния до предметов за счет закона перспективы или за счет указанного динамического эффекта глубины достаточно одного глаза. Однако природа снабдила человека двумя глазами, чтобы он имел максимальное ощущение объемности за счет восприятия одновременно двух различных ракурсов трехмерных сцен. Это самое мощное средство получения эффекта объемности из всех других свойств зрения человека. Однако для реализации максимальной достоверности воспроизведения сцены желательно использование всех имеющихся свойств.

Свойство бинокулярного зрения человека таково, что горизонтальный пространственный разнос (параллакс P) между парой точек на плоскости (при условии их раздельного наблюдения глазами) вызывает у человека ощущение восприятия некоторой единой точки, расположенной за плоскостью или перед ней, в зависимости от геометрии наблюдения данных точек (рис. 1). Стереоскопическое положение точки можно получить за экраном с помощью параллакса P^- отрицательного знака и перед экраном с помощью параллакса P^+ положительного знака.

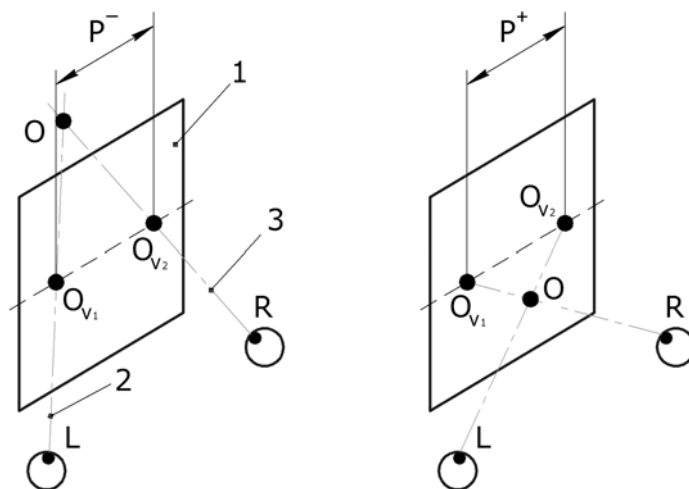


Рис. 1. Принцип получения стереоскопического изображения точки

Заключение

Использование стереоскопических и квазиостереоскопических позволяет получить весьма четкое впечатление от объемности наблюдаемой сцены. Это может найти широкое применение во многих областях науки, использующих пространственные построения. При условии присутствия пространственной перспективы в исходном моноскопическом изображении, а также наличии динамизма в сцене полученное квазистереоскопическое изображение будет практически мало отличимое от настоящего стереоизображения, если реализуемый метод преобразования изображения эффективно использует природные свойства зрения человека, отвечающие за восприятие глубины наблюдаемой сцены.

Литература

1. <<http://www.arbyte.ru/pdf/arbyteviziosapr9.pdf>>
2. Starks M. Method for producing a synthesized stereoscopic image. / Патент США № 6108005, НКИ 345/419, МКИ G06T 015/10, опублик. 22.08.2000.
3. Sebastian Knorr, Thomas Sikora. Modular Scheme for 2D/3D Conversion of TV Broadcast. / Communication Systems Group Technische Universitaet Berlin Einsteinufer 17, Berlin, Germany, 2006.
4. Ежов В.А. Способ получения объемного телевизионного изображения. / Патент РФ № 2113771, МКИ H04N 15/00, 20.06.98.

КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ ОРГАНОВ ЧЕЛОВЕКА

Н.Г. Рущенко, В.Т. Тозик, А.В. Меженин

В статье рассматриваются вопросы получения трехмерных моделей органов человека на основе томографических данных. Обсуждается один из методов — вычисление Фурье-образа проекций.

Введение

В настоящее время 3D-реконструкция внутренних органов человека на основе рентгенографии, ультразвуковых исследований, компьютерной и магнитно-резонансной томография получает все большее распространение и является актуальной задачей. Применение этих современных методов исследования позволяет врачу рассмотреть и оценить состояние органов и организма в целом. При этом, к примеру, при постановке вопроса о хирургической операции врач может реально увидеть оперируемый орган на экране компьютера и оценить объем хирургического вмешательства.

На кафедре ИКГ совместно с научно-исследовательским институтом им. Г.И. Турнера проводятся исследования в этой области.

Рассмотрим этапы получения трехмерных компьютерных моделей.

Первый этап — получение исходной морфометрической информации. Для этого используются клинико-инструментальные методы исследования (рентгенография, ультразвуковое исследование, компьютерная и магнитно-резонансная томография и др.). Полученные в результате проведенного исследования количественные данные используются на втором и третьем этапах для создания компьютерных моделей, имеющих установленные при морфометрии параметры и расположенные в виртуальном пространстве.

Второй этап создания компьютерной модели заключался в построении плоскостной двухмерной модели органа человека в трех проекциях.

Третий этап — трехмерная реконструкция и создание объемной компьютерной модели.

Второй и третий этапы планируется выполнить с использованием специально разработанного программного обеспечения и системы трехмерного моделирования 3ds max [1].

Созданные компьютерные модели могут быть продемонстрированы на дисплее, повернуты в любой плоскости и распечатаны на принтере в виде фотографий. Используя возможности программы 3ds max, можно перемещать компьютерные модели в виртуальном пространстве, что позволяет сдвинуть или удалить отдельные элементы модели для изучения особенностей топографии области. Кроме этого, планируется, используя систему инверсной кинематики, моделировать поведение элементов позвоночника при его лечении.

Трехмерные компьютерные модели могут быть полезны для повышения эффективности рентгенодиагностических методик, при планировании хирургических вмешательств. Кроме этого, возможно их использование как наглядных пособий не только в учебном процессе, но и различных научных исследованиях.

Использование преобразования Фурье для 3D-реконструкция объектов

Геометрия исходных данных (2D-проекций) при исследовании объектов томографическими методами [2] может быть получена расположением зоны 2D-проекций по образующим к цилиндрической поверхности, на осевой линии которой расположен исследуемый объект. В диаметрально противоположных направлениях позиционируется

источник излучения. Рассмотрим математический аппарат, который может быть использован в этом случае [3].

В каждой проекции регистрируемая интенсивность поля излучения – $I(x, y) \Rightarrow I(\vec{r})$. Для реконструкции в первом приближении выбирается функция в виде

$$D_z(x, y) = \int_0^L \mu(x; y; z) dz, \text{ здесь } D_z(x, y) - \text{двумерная функция проекции, } L -$$

толщина объекта в направлении просвечивания, $\mu(x; y; z)$ – линейный (массовый) коэффициент ослабления по объему контролируемого образца. Если плотности внутренних локальных объектов постоянны, то последнее соотношение можно упростить:

$$D_z(x_i; y_i) = \mu_0 L + \sum_k (\mu_1 - \mu_0)_k l(x_i; y_i). \text{ Здесь } \mu_0 - \text{коэффициент ослабления материала изделия, } \mu_1 - \text{коэффициент ослабления материала локального объекта, } l(x_i; y_i) -$$

размер локального объекта в направлении просвечивания, k – их число в направлении просвечивания. Нормализованные функции проекций –

$$g(p) = \int_L f(r, \theta) \delta(p - \vec{n} \vec{r}) dl, \text{ в том числе и одномерные } g(r; \theta), g(p), \text{ «вычисляемые»}$$

из двумерных, должны удовлетворять условию

$$\Re [g(p)] = \int_0^{2\pi} d\theta \int_L f(r, \theta) \delta(p - \vec{n} \vec{r}) dl.$$

Точность реконструкции формы поверхности в этом варианте определяется точностью восстановления границ сечений, поэтому разработан следующий способ реконструкции сечений и контуров исследуемых объектов.

- Задается формат сечения. Форматы изображений локальных объектов выбираются в пределах (64×64), (256×256) элементов. Требуемое число проекций определяется конечным выбранным форматом изображения. Этот набор $g_{\theta_i}(p)$ дополняется набором $g_{\theta_i+\pi}(p)$. Недостающее количество проекций доопределяется, методами интерполяции.
- Для данного сечения вычисляются Фурье-образы проекций при числе отсчетов в каждой проекции, в несколько (3–6) раз превышающих линейный формат матрицы изображения сечения. В соответствии с теоремой о сечениях формируется «двумерный Фурье-спектр».
- По кольцевым гармоникам производится интерполяции в зонах отсутствия отсчетов. Так как спектры реальных проекций содержат дополнительные отсчеты, то процедуру интерполяции можно оптимизировать, используя зоны дополнительных отсчетов, и таким образом оптимизировать двумерный спектр Фурье – «выгладить» его.
- Наборы одномерных спектров проекций пересчитываются в функции проекций, оптимизируется ядро алгоритма свертки. Для данного класса задач ядро оказывается достаточно «низкочастотным», в общем случае его Фурье-образ в области главного максимума имеет вид

$$G(\omega) = \frac{a(2\pi)^2 e^{-2\pi\omega} \cos(\omega^2) \sin(\omega^2)}{1 + \omega^2},$$

здесь a – константа, выбираемая в пределах (0,9–2,4).

- После нормализации всех функций проекций (реальных и вычисленных) для данного сечения производится реконструкция каждого сечения с помощью стандартной

процедуры – сверточного алгоритма, включающего вычисление одномерных сверток проекций с выбранным ядром и процедуру обратного проецирования.

Таким образом, формируется заданное количество сечений, при этом все вычислительные процедуры реализуют только одномерные преобразования: вычисление прямых, обратных преобразований Фурье от функций проекций, интерполяция по кольцевым гармоникам в двумерном Фурье спектре, операция свертки. Это сокращает время реконструкции, алгоритмы могут быть реализованы на стандартных ПК. В то же время фактически оказывается вычисленным 3D-массив, определяющий конфигурацию локального объекта и его ориентацию в пространстве.

Заключение

Рассмотренный математический аппарат положен в основу разрабатываемого программного обеспечения, которое в настоящее время находится в стадии отладки.

Литература

1. 3ds Max 9: трехмерное моделирование и анимация / В.Т. Тозик, А.В. Меженин. – СПб.: БХВ-Петербург, 2007. 1056 с.: ил.
2. Рущенко Н.Г. Исследование и разработка методов решения задачи синтеза высокооднородного магнитного поля в МР- томографе. Автореферат дисс. канд.техн.наук – СПб: ИТМО,2004.
3. Филонин О.В., Шадрунов А.А. Цифровые методы обработки в компьютерной томографии / Тр. межд. симп. «Надежность и качество». Пенза, 2004. С. 257–261.

Бердникова Елена Александровна – ассистент кафедры ТПО, программист ЦДО
Береговой Михаил Викторович – аспирант кафедры БИТ
Васильев Владимир Николаевич – доктор технических наук, профессор, ректор
Вашенков Олег Евгеньевич – студент кафедры ПИКО, программист ЦДО
Волкова Анастасия Алексеевна – аспирант кафедры КОТ, программист ЦДО
Гордеева Наталья Олеговна – ассистент кафедры ТПО
Горлушкина Наталья Николаевна – кандидат технических наук, доцент кафедры ТПО
Дергачев Андрей Михайлович – старший преподаватель кафедры ВТ.
Дульнев Геннадий Николаевич – доктор технических наук, профессор кафедры КТФиЭМ
Зайцев Олег Евгеньевич – аспирант кафедры БИТ
Зверев Александр Олегович – аспирант кафедры ВТ
Зыков Анатолий Геннадьевич – кандидат технических наук, доцент кафедры ИиПМ
Кессель Сергей Владимирович – аспирант кафедры ТПО
Ковязин Рустам Раисович – магистр математики, ведущий программист ООО «ЛМТ»
Ковязина Динара Раисовна – студентка кафедры ВТ
Корнев Дмитрий Александрович – студент кафедры ТПО
Лапшина Ирина Анатольевна – кандидат технических наук, доцент
Лисицына Любовь Сергеевна – кандидат технических наук, заведующая кафедрой КОТ
Лукичев Александр Николаевич – аспирант кафедры ВТ
Лукьянов Николай Михайлович – аспирант кафедры ВТ
Любимов Александр Вилиевич – кандидат технических наук, доцент кафедры БИТ
Лямин Андрей Владимирович – кандидат технических наук, доцент, директор ЦДО
Медведева Татьяна Сергеевна – студентка кафедры ТПО
Меженин Александр Владимирович – кандидат технических наук, доцент кафедры ИКТ
Немолочнов Олег Фомич – доктор технических наук, профессор, заведующий кафедрой ИиПМ
Неронов Юрий Ильич – доктор физико-математических наук, профессор
Нестерук Геннадий Филиппович – доктор технических наук, профессор кафедры БИТ
Нестерук Филипп Геннадьевич – кандидат технических наук, доцент кафедры БИТ
Овчаров Денис Викторович – студент кафедры ВТ
Окулевич Владимир Викентьевич – кандидат технических наук, ведущий инженер ООО Siemens

Осовецкий Леонид Георгиевич – доктор технических наук, профессор, заведующий кафедрой БИТ

Петров Евгений Владимирович – студент

Петров Кирилл Владимирович – аспирант кафедры ИиПМ

Плешкова Мария Витальевна – аспирант кафедры КОТ, программист ЦДО

Поляков Владимир Иванович – кандидат технических наук, доцент кафедры ВТ

Потеев Михаил Иванович – кандидат технических наук, профессор, заведующий кафедрой ТПО

Птицын Алексей Владимирович – кандидат технических наук, доцент кафедры БИТ

Рущенко Нина Геннадьевна – кандидат технических наук, доцент кафедры ИКГ

Сизиков Валерий Сергеевич – доктор технических наук, профессор кафедры ИТиКТ

Скалецкая Наталья Дмитриевна – аспирант кафедры ИТиКТ

Скорубский Владимир Иванович – кандидат технических наук, доцент кафедры ВТ

Соколов Дмитрий Александрович – аспирант кафедры ИНС

Стражмейстер Инна Борисовна – старший преподаватель кафедры ТИО

Терещенко Евгения Николаевна – аспирантка кафедры ИТиКТ

Тимченко Борис Дмитриевич – кандидат технических наук, доцент кафедры ВТ

Тозик Вячеслав Трофимович – кандидат технических наук, доцент кафедры ИКГ

Хлопотов Максим Валерьевич – ассистент кафедры ТПО

Храпов Сергей Викторович – магистрант кафедры ВТ

Чежин Михаил Сергеевич – кандидат технических наук, доцент кафедры СУИ, заместитель директора ЦДО

Шедько Ольга Геннадьевна – аспирантка кафедры ВТ

1. ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ОБРАЗОВАНИИ	3
Потеев М.И., Хлопотов М.В. Методика подготовки будущих инженеров специальности «Информационные технологии в образовании».....	3
Лапшина И.А., Медведева Т.С., Корнев Д.А. Структура, наполнение и программная реализация интерактивного сайта газеты вуза	11
Горлушкина Н.Н., Гордеева Н.О. Методика обучения проектированию цифровых образовательных ресурсов	16
Бердникова Е.А. Возможность и перспективы использования ресурсов системы дистанционного обучения СПбГУ ИТМО для функционирования виртуального представительства системы открытого образования.....	20
Кессель С.В. Проблемы подготовки профессиональных консультантов в области информационных технологий и пути их решения	23
Дульнев Г.Н., Стражмейстер И.Б. Энергоинформационные потоки при обучении и методы их исследования.....	27
2. СЕТИ ЭВМ И ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ	32
Лукьянов Н.М., Дергачев А.М. Ложные вычислительные системы для исследования и отвлечения атак	32
Зверев А.О. Тимченко Б.Д. SDM-модель клиент-серверные системы	39
Храпов С.В. Использование специализированной системы моделирования NS-2 для исследования телекоммуникационных сетей	46
Храпов С.В. Формирование связанных топологий специализированных сетей Bluetooth Scatternets.....	52
3. ИНФОРМАЦИОННО-УПРАВЛЯЮЩИЕ СИСТЕМЫ	58
Лукичев А.Н. Вычислительные механизмы как средство проектирования встроенных систем	58
Петров Е.В. Обновление программного обеспечения в распределенных управляющих системах.....	65
Ковязина Д.Р. Синхронизация времени в распределенных информационно-управляющих системах	71
Ковязин Р.Р. Разработка информационно-управляющих систем как актуализация целевой задачи	79
Шедько О.Г., Окулевич В.В. Поиск программных компонент для повторного использования с помощью автоматизированных систем доказательств теорем	86
Овчаров Д.В., Скорубский В.И. Средства тестирования печатных плат с использованием технологии JTAG	91
4. ТЕОРИЯ И ТЕХНОЛОГИЯ ПРОГРАММИРОВАНИЯ И ЗАЩИТЫ ИНФОРМАЦИИ	97
Зайцев О.Е., Любимов А.В. Формальное моделирование «общих критериев» как инструмент внедрения и продвижения стандарта ГОСТ Р ИСО/МЭК 15408.....	97
Нестерук Г.Ф., Нестерук Ф.Г., Осовецкий Л.Г. Обеспечение избыточности информационных полей адаптивных классификаторов для систем защиты информации	105

Птицын А.В. Модели процесса обнаружения угроз при доступе к ресурсам кластерной системы	110
Осовецкий Л.Г., Береговой М.В. Структурное моделирование конкурентной борьбы	116
Немолочнов О.Ф., Зыков А.Г., Поляков В.И., Осовецкий Л.Г., Петров К.В. Методы тестирования вычислительных процессов	121
5. НОВЫЕ ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ	126
Васильев В.Н., Лисицына Л.С., Лямин А.В. Технология проведения ЕГЭ по информатике в компьютерной форме	126.
Лисицына Л.С. Сетевое сообщество методического Интернет- центра: современное состояние проекта и перспективы развития	143
Васильев В.Н., Лямин А.В., Чежин М.С. Система дистанционного обучения второго поколения	148
Вашенков О.Е., Волкова А.А., Лямин А.В. Примеры реализации сетевых виртуальных лабораторий в среде системы дистанционного обучения	157
Лямин А.В., Плешкова М.В. Разработка среды программирования структуры электронных курсов	164
Свечников С.В. Подход к созданию средств тематической категоризации Интернет-ресурсов	167
Силаев А.В. Некоторые вопросы открытой архитектуры вертикальных образовательных порталов	175
6. ИЗМЕРИТЕЛЬНЫЕ ТЕХНОЛОГИИ И КОМПЬЮТЕРНАЯ ТОМОГРАФИЯ	181
Терещенко Е.Н. Развитие неинвазивных методов оценки входной функции по ПЭТ-изображениям	181
Скалецкая Н.Д. Теоретико-групповые методы формирования шкал коррекции пространственных искажений изображений в томографии	185
Неронов Ю.И., Сизиков В.С., Соколов Д.Ю. Сравнение трех методов синтеза однородного поля перманентного магнита в МР-томографии	190
7. ИНЖЕНЕРНАЯ И КОМПЬЮТЕРНАЯ ГРАФИКА	203
Меженин А.В., Тозик В.Т. Реконструкция трехмерных моделей по растровым изображениям	203
Тозик В.Т. Получение квазистереоскопических изображений	208
Рущенко Н.Г., Тозик В.Т., Меженин А.В. Компьютерное моделирование органов человека	211
НАШИ АВТОРЫ	214

Научно-технический вестник СПбГУ ИТМО. Выпуск 45. Информационные технологии / Главный редактор д.т.н., проф. В.Н. Васильев. – СПб: СПбГУ ИТМО, 2007. 218 с.

НАУЧНО-ТЕХНИЧЕСКИЙ ВЕСТНИК СПбГУ ИТМО
Выпуск 45

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

Главный редактор
доктор технических наук, профессор
В.Н. Васильев

Дизайн обложки В.А. Петров, А.А. Колокольников
Редакционно-издательский отдел СПбГУ ИТМО
Зав. РИО Н.Ф. Гусарова

Лицензия ИД № 00408 от 05.11.99.

Подписано в печать 15.12.07.

Заказ 1100. Тираж 100 экз.