

СОДЕРЖАНИЕ

| | |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| Гуров В.С., Гостин А.М., Корячко В.П., Митрошин А.А., Таганов Р.А., Чернышев С.В. Структурное и функциональное построение автоматизированной системы дистанционного обучения и повышения квалификации специалистов в области создания и использования спутниковых навигационных систем | 6 |
| Корячко В.П., Таганов А.И., Таганов Р.А., Чернышев С.В. Многокритериальный анализ и выбор платформы для реализации виртуальной образовательной среды специального назначения..... | 16 |
| Корячко В.П., Таганов Р.А., Чернышев С.В. Методологические принципы построения виртуальных учебных лабораторий | 28 |
| Аникеев С.В., Дубинина В.Л., Маркин А.В. О средствах генерации отчетов..... | 33 |
| Аникеев С.В., Дубинина В.Л., Маркин А.В. О технологиях создания отчетов в информационных системах | 41 |
| Бакулев А.В., Митрошин А.А. О возможности использования существующих ресурсов Internet при создании Web-портала системы дистанционного обучения ГЛОНАСС | 48 |
| Баскакова О.М. Использование современных информационных технологий дистанционного обучения на примере системы MOODLE | 53 |
| Гостин А.М., Самохина Н.В., Фокина А.А. Принципы построения системы дистанционного обучения по подготовке специалистов в области использования ГЛОНАСС..... | 55 |
| Гостин А.М., Фокина А.А., Самохина Н.В. Структурно-функциональная схема АСДО | 62 |
| Гостин А.М., Таганов Р.А., Чернышев С.В. Принципы разработки виртуальных интеллектуальных тренажеров..... | 69 |
| Дондик Е.М., Вавилова О.С. Система анализа искаженных фонем | 75 |

| | |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| Зорина М.А., Небыкова Е.Д. Разработка модуля обработки обращения клиента для банковской информационной системы с использованием CRM-стратегий | 81 |
| Карманов П.В. Дискретный фазовращатель..... | 83 |
| Карманов П.В. Цифровое моделирование импульсного квадратурно-фазового дискриминатора при случайном входном сигнале | 87 |
| Кий Д.В., Некротов К.Е., Ручкин В.Н., Фулин В.А. Управление образовательными ресурсами посредством современных информационных технологий | 93 |
| Козлов М.А., Кравчук Н.В. Разработка веб-сервиса для локального просмотра контента веб-сайтов..... | 96 |
| Крошлин А.В. Предварительная подготовка массивов данных для кластеризации с использованием технологии TEXT MINING | 99 |
| Курдюмова М.В. Способы увеличения производительности веб-приложений | 101 |
| Митрошин А.А., Новиков А.А., Таганов Р.А., Чернышев С.В. Возможность использования метода СОСОМО при оценке себестоимости разработки программного обеспечения АСДО по использованию ГЛОНАСС | 104 |
| Подгорнова Н.А. Применение вейвлет-анализа и вейвлет-преобразования для исследования и обработки временных рядов денежных потоков предприятия..... | 115 |
| Розумеенко М.С. Семантические ошибки в SQL запросах и способы их возникновения | 122 |
| Рюмин Р.М. Оптимизация тестирования работы компьютерной сети..... | 126 |
| Скоробогатова Н.Е., Дондик Е.М. Системы распознавания дактильной речи..... | 127 |

| | |
|-----------------------------------------------------------------------------------------------------------------------------------------------|-----|
| Таганов А.И. Методы представления рисков проекта на основе структурно-символьных моделей..... | 133 |
| Таганов А.И. Метод синтеза логико-алгебраических моделей рисков проекта | 141 |
| Фомин С.Н., Перепелкин Д.А. Разработка комплекса программных средств моделирования алгоритмов маршрутизации в корпоративных сетях..... | 151 |
| Харченко А.С., Минаев Ю.М., Ручкин В.Н. Система управления историческими знаниями..... | 155 |
| Хруничев Р.В. Электронные издания: терминология и классификация | 158 |

**В.С. ГУРОВ, А.М. ГОСТИН, В.П. КОРЯЧКО,
А.А. МИТРОШИН, А.И. ТАГАНОВ, С.В. ЧЕРНЫШЕВ**
Рязанский государственный радиотехнический университет

**СТРУКТУРНОЕ И ФУНКЦИОНАЛЬНОЕ ПОСТРОЕНИЕ
АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ ДИСТАНЦИОННОГО
ОБУЧЕНИЯ И ПОВЫШЕНИЯ КВАЛИФИКАЦИИ
СПЕЦИАЛИСТОВ В ОБЛАСТИ СОЗДАНИЯ
И ИСПОЛЬЗОВАНИЯ СПУТНИКОВЫХ
НАВИГАЦИОННЫХ СИСТЕМ**

Рассматривается структурное и функциональное построение разрабатываемой автоматизированной системы дистанционного обучения и повышения квалификации специалистов в области создания и использования спутниковых навигационных систем.

Введение

Глобальная навигационная система ГЛОНАСС, созданная однажды преимущественно для нужд оборонного значения, переживает сейчас второе рождение, которое заключается в реализации государственных планов по эффективному использованию возможностей космических средств навигации во многих отечественных отраслях и в том числе в геодезии, строительстве и на транспорте [1].

Реализация этих планов тесно сопряжена с решением другой государственной проблемы, определяющейся в необходимости подготовки и повышении квалификации широкого круга специалистов в указанных отраслях с целью передачи им знаний и мирового опыта по эффективному и результативному использованию больших возможностей навигационной системы ГЛОНАСС.

Практическая реализация в Российской Федерации образовательных планов осуществляется сейчас посредством организации специальных конкурсных мероприятий в рамках федеральной целевой программы «Глобальная навигационная система». Одним из таких мероприятий является проведение ОКР «Квалификация», составную часть которой по созданию автоматизированной системы дистанционного обучения (АСДО) на конкурсной и контрактной основе выполняет Рязанский государственный радиотехнический университет.

В соответствии с техническим заданием ОКР «Квалификация» основной целью создания АСДО [2] является обеспечение процесса подготовки и повышения квалификации специалистов в области создания и использования спутниковых навигационных систем (СНС) для субъектов Российской Федерации, заинтересованных министерств и

ведомств, коммерческих структур, некоммерческих организаций и массовых потребителей путём использования автоматизированных средств и методов дистанционного обучения.

1. Основные функциональные задачи АСДО

Разрабатываемая автоматизированная система дистанционного обучения предназначена для решения важных функциональных задач в тесном взаимодействии со смежным информационно-аналитическим комплексом (ИАК) и комплексом научно-технического и методического обеспечения (КНТМО). Основными планируемыми функциями АСДО являются:

- получение исходных данных от смежных систем (ИАК, КНТМО) для текущего оперативного планирования процесса подготовки специалистов в области спутниковой навигации;
- регистрация и распределение прав доступа к ресурсам и функциям АСДО преподавателей и администраторов;
- формирование электронных учебных курсов в составе интерактивных электронных обучающих систем (ИЭОС);
- ведение каталогов учебных ресурсов;
- формирование групп обучаемых и назначения необходимых преподавателей для групп;
- планирование учебного процесса;
- обеспечение информационного взаимодействия со слушателями;
- проведение промежуточного и итогового тестирования;
- ведение статистики активных действий слушателей;
- проведение занятий по дистанционной технологии;
- формирование отчетов по обучению и статистике и др.

Для реализации в АСДО указанных функций проведены исследования и анализ существующих коммерческих и свободно распространяемых систем дистанционного обучения и разработан системный проект АСДО, учитывающий передовой отечественный и мировой опыт создания подобных систем.

2. Структурное построение АСДО

Разработанная структурная схема построения АСДО включает в себя следующие компоненты (рис. 1):

- комплекс технических средств АСДО;
- специальное программное обеспечение АСДО;
- информационное обеспечение АСДО;

- интерактивные электронные обучающие системы (ИЭОС). При этом комплекс технических средств АСДО содержит:
- серверные технические средства;
- технические средства рабочих мест (РМ) на основе ПЭВМ;
- общее программное обеспечение;
- телекоммуникационные средства и средства защиты информации.

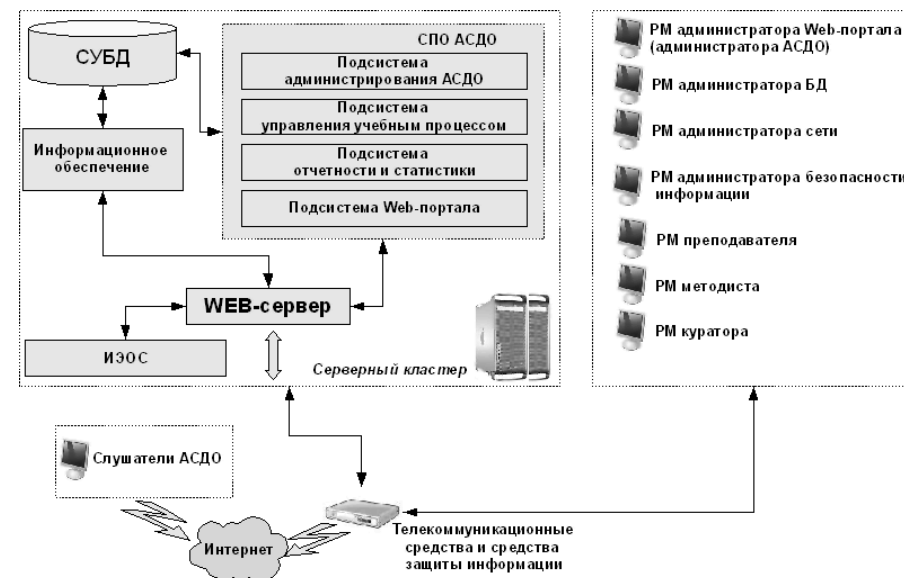


Рис. 1. Структурная схема АСДО

В планируемый и проектно-обоснованный состав специального программного обеспечения (СПО) АСДО вошли:

- комплекс программ администрирования АСДО;
- комплекс программ управления учебным процессом;
- комплекс программ отчетности и статистики;
- комплекс программ управления Web-порталом.

При этом в представленном составе СПО подсистема администрирования АСДО состоит из программных средств (ПС) управления учетными данными пользователей АСДО, реализующих задачи АСДО в части администрирования.

Подсистема управления учебным процессом состоит уже из ПС и модулей, реализующих задачи АСДО в части управления учебным процессом:

- ПС планирования учебного процесса;
- ПС администрирования учебного процесса;
- ПС мониторинга обучения;
- ПС информационного взаимодействия с ИАК.

Подсистема отчетности и статистики состоит из ПС и модулей, реализующих задачи АСДО в части формирования отчетности:

- ПС формирования и отправки статистики для ИАК;
- ПС формирования отчетов по обучению.

Подсистема Web-портала АСДО состоит из программных средств и модулей, реализующих задачи интеграции и взаимодействия компонентов АСДО:

- ПС интеграции компонентов АСДО;
- ПС информационного взаимодействия пользователей в процессе дистанционного обучения;
- ПС поиска и печати учебных материалов в электронных учебных курсах (ЭУК);
- ПС формирования интерфейса.
- В состав информационного обеспечения АСДО также входят:
- базы данных (БД) нормативно-справочной информации в области создания и использования спутниковых навигационных систем;
- БД всей системы АСДО.

В свою очередь интерактивные электронные обучающие системы (ИЭОС) состоят из компонентов:

- - программных средств;
- - информационного обеспечения.

При этом программные средства ИЭОС включают в себя:

- ПС проигрывателя ЭУК;
- ПС создания комплекса автономного обучения;
- ПС каталогизированного хранения ЭУК;
- ПС редактирования ЭУК, состоящие из: редактора теоретических материалов ЭУК; редактора тестов; редактора упражнений;
- ПС информационного взаимодействия с комплексом научно-технического и методического обеспечения.

Информационное обеспечение ИЭОС содержит базу данных ИЭОС.

В процессе разработки и анализа структуры АСДО были определены четыре категории пользователей и их роли в системе:

- пользователи АСДО - обучаемые (слушатели);
- пользователи АСДО - преподаватели (обучающие);
- пользователи АСДО - работники центра дистанционного обучения (ЦДО): преподаватели; разработчики курсов и ИЭОС; методисты; публикатор WEB-портала; редактор WEB-портала; учебно-вспомогательный персонал;
- эксплуатационный персонал АСДО - работники ЦДО: администраторы АСДО; администраторы баз данных АСДО; администраторы вычислительной сети АСДО; администраторы безопасности информации (персонал службы обеспечения безопасности информации) АСДО; администратор WEB-портала.

Для реализации указанной структуры АСДО был проанализирован мировой и отечественный опыт в области создания систем управления обучением и систем дистанционного обучения [3-6]. Главным критерием выбора прототипа стало соответствие выбираемой системы дистанционного обучения запланированной структуре.

Среди открыто распространяемых СДО по ряду параметров лидирует СДО Moodle [5]. Среди коммерческих систем дистанционного обучения лидирует Training Ware ЗАО «Корпоративные Системы Обучения», которая была выбрана в качестве прототипа для АСДО [6].

3. Процесс функционирования АСДО

Процесс функционирования АСДО может быть представлен блок-схемой, отражающей алгоритм взаимодействия отдельных модулей АСДО и внешних систем (рис. 2):

1. ИАК после получения заявки на обучение составляет примерный график обучения и принимает решение о номенклатуре курса повышения квалификации, анализирует данные, предоставляемые АСДО об организационных ресурсах АСДО (количество преподавателей, загрузка системы) и номенклатуре курсов, функционирующих в АСДО. Специалисты ИАК анализируют необходимость привлечения дополнительного учебно-вспомогательного состава (УВП) и профессорско-преподавательского состава (ППС) и привлечения учебно-методических комплексов (УМК) КНТМО для обеспечения учебного процесса.

2. ИАК формирует на основе данных анализа примерный график обучения (сроки) и кодификатор курса обучения, данные о численности потенциальных слушателей, при необходимости данные о необходимости привлечения дополнительного УВП и ППС и их численности и квалификации, при необходимости запрос к КНТМО на формирование курса, пересылает данные в АСДО и КНТМО.

3. АСДО на основании данных, полученных из ИАК, посредством персонала ЦДО заключает договора на обучение с потенциальными слушателями или организацией. Администратор АСДО регистрирует их в АСДО, заключает договора с УВП и ППС, а также администратор АСДО регистрирует их в системе на соответствующие роли.

4. В административную подсистему АСДО пересылается из КНТМО курс обучения в формате SCORM. Администратор АСДО (разработчик курсов) импортирует курс в АСДО, включая его в состав ИЭОС, присваивает ему номенклатуру в соответствии с классификаторами системы.

5. Администратор АСДО формирует группы обучения, назначает кураторов для каждой группы, преподавателей.

6. Для каждой группы слушателей для поддержки курса формируются средства коммуникации, назначаются модераторы средств коммуникаций;

7. Для каждой группы слушателей на основе учебного плана из УМК КНТМО и графика обучения из ИАК формируется расписание занятий;

8. Куратор группы с помощью сформированных администратором средств коммуникации оповещает слушателей о начале обучения;

9. Слушатели повышения курса повышения квалификации с помощью ИЭОС взаимодействуют с дистанционными учебными курсами, осваивают элементы курса, выполняют промежуточные и итоговые контрольные задания, упражнения;

10. Подсистема статистики АСДО фиксирует действия пользователей, формирует журнал работы системы, отчеты о результатах обучения;

11. Отчеты о результатах обучения слушателей и другая статистическая информация передается в ЦДО и ИАК;

12. На основе данных о результатах обучения и критериев успешности освоения курса из КНТМО административный персонал АСДО формирует сертификат об окончании курсов повышения квалификации.

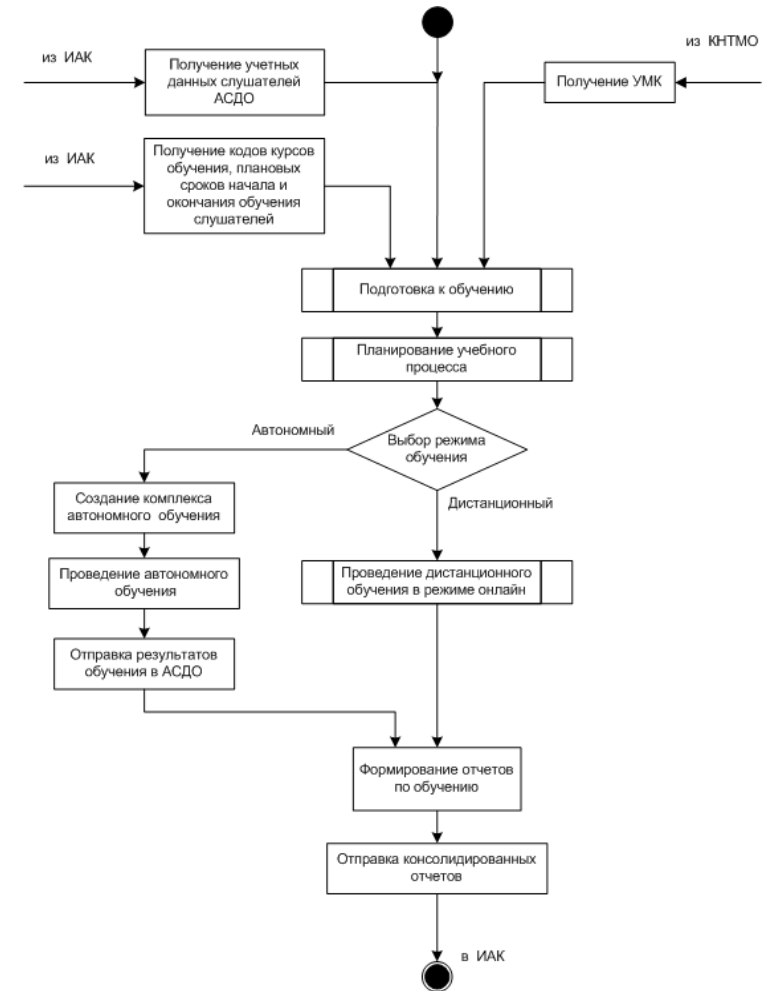


Рис. 2. Функциональное построение АСДО

Взаимодействие слушателя с АСДО организуется с помощью ИЭОС посредством интерфейса Web-портала. При этом управление процессом обучения осуществляется на основе полномочий (статуса) пользователей. В АСДО определены следующие статусы пользователей:

- *куратор* - выполняет планирование и контроль результатов обучения;
- *преподаватель (тьютор)* – выполняет разработку и адаптацию учебно-методического обеспечения, подготовку и проведение занятий, консультационную поддержку слушателей;
- *слушатель* – проходит обучение;
- *администратор* – выполняет информационное наполнение, администрирование и управление настройками программного комплекса.

Для организационного решения задачи дистанционного обучения планируется создание центра дистанционного обучения, который может включать в себя компьютерный учебный класс с локальной вычислительной сетью, подключенный к серверному узлу АСДО. Этот класс может быть использован для проведения семинаров, очных консультаций с преподавателями, итоговых зачетов и экзаменов по образовательным программам.

4. Процедура проведения обучения средствами АСДО

Структурная схема процедуры проведения дистанционного обучения с использованием средств АСДО представлена на рис. 3. Из схемы следует, что обучение в АСДО проводится с помощью электронных учебных курсов (ЭУК), содержание которых формируется в КНТМО, а отдельные модули ЭУК могут реализовываться с помощью ИЭОС АСДО.

При обучении с помощью АСДО обеспечивается многопользовательский режим обработки информации с разграничением прав доступа пользователей АСДО к ресурсам АСДО. Взаимодействие пользователей с ресурсами АСДО осуществляется с использованием сети международного информационного обмена – сети Internet.

Встроенный редактор курсов дает возможность разработки дистанционного учебного курса с использованием графического интерфейса пользователя, при этом обеспечивается:

- формирование трехуровневой структуры представления учебного курса: модуль, занятие, цель обучения;
- разработка и редактирование содержательной части цели обучения;
- разработка и редактирование упражнений;
- разработка тестовых заданий, обеспечивающих проведение контроля знаний обучаемых.

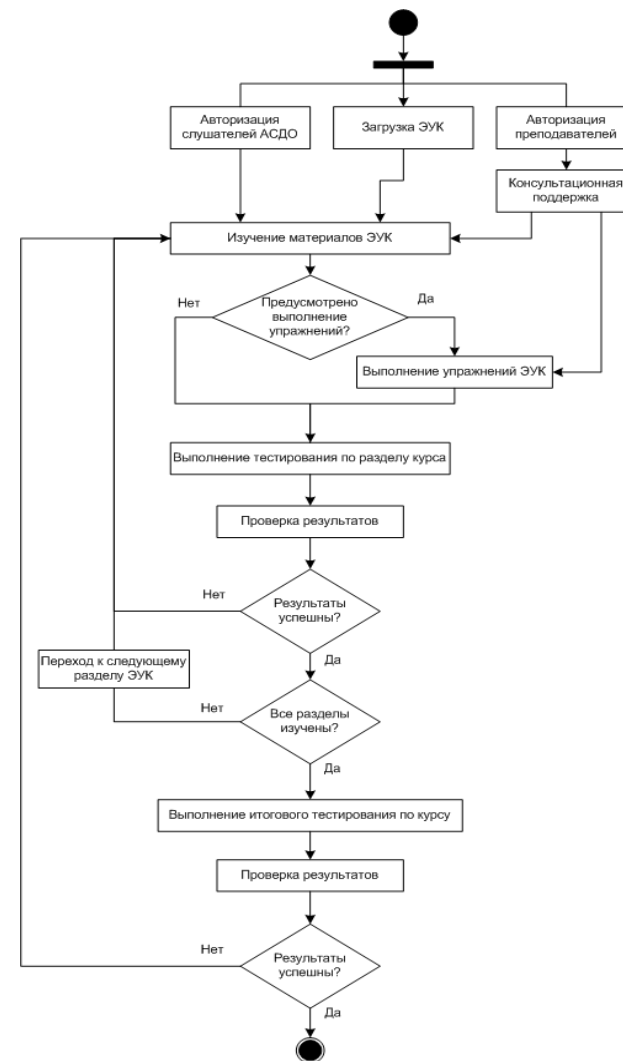


Рис. 3. Структурная схема процедуры проведения дистанционного обучения в АСДО

Содержательная часть дистанционного учебного курса может содержать: текстовую, графическую, аудио информацию, поддерживаемую базовым программным обеспечением АРМ обучаемого.

АСДО должен иметь встроенный конструктор упражнений и обеспечивать возможность использования базовых учебно-методических средств (БУМС) КНТМО в учебном процессе.

Редактор упражнений должен обеспечивать возможность:

- автоматизированной фиксации корректности выполнения действий при выполнении упражнения;
- использования графических объектов, моделирующих функционал изучаемого приложения при выполнении упражнения;
- отображения пошаговых инструкций по выполнению упражнения;
- настройки параметров выполнения упражнения (ограничения по времени выполнения упражнения и т.д.).

Упражнение разрабатывается и включается в состав дистанционного учебного курса ИЭОС в виде законченного модуля или ссылки на БУМС КНТМО. Упражнение должно поддерживать режимы выполнения сценариев и выполняться базовым программным обеспечением АРМ пользователя.

Телекоммуникационное взаимодействие с пользователями АСДО осуществляется сетевыми средствами и средствами территориальной сети Интернет.

Заключение

В результате разработки системного проекта АСДО обоснованы модели структурного и функционального построения АСДО, отвечающие современным требованиям по назначению и сервису. Разработаны общие алгоритмы и структурные схемы процедур дистанционного обучения, а также определены форматы обмена данными со смежными системами дистанционного обучения в процессе реализации образовательных программ по спутниковой навигации на базе системы ГЛОНАСС.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. <http://www.glonass-ianc.rsa.ru>.
2. Техническое задание на ОКР «Разработка автоматизированной системы дистанционного обучения подготовки и повышения квалификации специалистов по спутниковой навигации». М.: ОАО «Российские космические системы», 2009.
3. Соловов А.В. Электронное обучение: проблематика, дидактика, технология. – Самара: «Новая техника», 2006. – 454 с.
4. Новый подход к инженерному образованию: теория и практика открытого доступа к распределенным информационным и техниче-

ским ресурсам. Под редакцией А.А.Полякова. М.: Центр-Пресс, 2000. – 238 с.

5. http://www.opentechnology.ru/info/moodle_about.mtd

6. <http://www.hr-portal.ru>

**В.П. КОРЯЧКО, А.И. ТАГАНОВ, Р.А. ТАГАНОВ,
С.В. ЧЕРНЫШЕВ**

Рязанский государственный радиотехнический университет

МНОГОКРИТЕРИАЛЬНЫЙ АНАЛИЗ И ВЫБОР ПЛАТФОРМЫ ДЛЯ РЕАЛИЗАЦИИ ВИРТУАЛЬНОЙ ОБРАЗОВАТЕЛЬНОЙ СРЕДЫ СПЕЦИАЛЬНОГО НАЗНАЧЕНИЯ

Рассматривается решение задачи многокритериального анализа и выбора альтернатив в условиях нечеткости.

Введение

Проблема выбора платформы, на которой будет построена виртуальная обучающая среда проекта «ГЛОНАСС-КВАЛИФИКАЦИЯ» является ключевой и этот выбор зависит от целого ряда критериев и факторов [1]. К основным критериям выбора средств организации электронного обучения можно отнести следующие [2,3]:

- *Функциональность.* Обозначает наличие в системе набора функций различного уровня, таких как форумы, чаты, анализ активности обучаемых, управление курсами и обучаемыми, а также другие;
- *Надежность.* Этот параметр характеризует удобство администрирования и простоту обновления контента на базе существующих шаблонов. Удобство управления и защита от внешних воздействий существенно влияют на отношение пользователей к системе и эффективность ее использования;
- *Стабильность.* Означает степень устойчивости работы системы по отношению к различным режимам работы и степени активности пользователей;
- *Стоимость.* Складывается из стоимости самой системы, а также из затрат на ее внедрение, разработку курсов и сопровождение, наличие или отсутствие ограничений по количеству лицензий на слушателей (студентов);
- *Наличие средств разработки контента.* Встроенный редактор учебного контента не только облегчает разработку курсов, но и позволяет интегрировать в едином представлении образовательные материалы различного назначения;

- **Поддержка SCORM.** Стандарт SCORM является международной основой обмена электронными курсами и отсутствие в системе его поддержки снижает мобильность и не позволяет создавать переносимые курсы;

- **Система проверки знаний.** Позволяет в режиме онлайн оценить знания учеников. Обычно такая система включает в себя тесты, задания и контроль активности обучаемых на форумах;

- **Удобство использования.** При выборе новой системы необходимо обеспечить удобство ее использования. Это важный параметр, поскольку потенциальные ученики никогда не станут использовать технологию, которая кажется громоздкой или создает трудности при навигации. Технология обучения должна быть интуитивно понятной. В учебном курсе должно быть просто найти меню помощи, должно быть легко переходить от одного раздела к другому и общаться с инструктором.

- **Модульность.** В современных системах ЭО курс может представлять собой набор микромодулей или блоков учебного материала, которые могут быть использованы в других курсах.

- **Обеспечение доступа.** Обучаемые не должны иметь препятствий для доступа к учебной программе, связанных их расположением во времени и пространстве, а также с возможными факторами, ограничивающими возможности обучаемых (ограниченные функции организма, ослабленное зрение). Также использование технологий «завтрашнего дня», которые поддерживаются ограниченным кругом программно-обеспечения, существенно снижает круг потенциальных пользователей.

- **100% мультимедийность.** Возможность использования в качестве контента не только текстовых, гипертекстовых и графических файлов, но и аудио, видео, gif- и flash-анимации, 3D-графики различных файловых форматов.

- **Масштабируемость и расширяемость.** Возможность расширения как круга слушателей обучаемых по СДО, так и добавления программ и курсов обучения и образования.

- **Перспективы развития платформы.** СДО должна быть развивающейся средой, должны выходить новые, улучшенные версии системы с поддержкой новых технологий, стандартов и средств.

- **Кросс-платформенность СДО.** В идеале система дистанционного обучения не должна быть привязана к какой-либо операционной системе или среде, как на серверном уровне, так и на уровне клиентских машин. Пользователи должны использовать стандартные средства без загрузки дополнительных модулей, программ и т.д.

- **Качество технической поддержки.** Возможность поддержки работоспособности, стабильности СДО, устранения ошибок и уязвимостей как с привлечением специалистов компании разработчика СДО, так и специалистами собственной службы поддержки организации.

- **Наличие (отсутствие) русской локализации продукта.** Локализованная версия продукта более дружелюбная как для администрирования, разработки курсов, так и для конечных потребителей образовательных услуг.

Указанная задача выбора платформы из множества альтернативных и доступных коммерческих систем по указанным критериям является достаточно сложной и по причине нечеткости и субъективности исходных экспертных данных по оценкам и требует привлечения для своего решения апробированных формализованных методов теории нечетких множеств [4,5].

1. Теоретический анализ

Рассмотрим метод принятия решений для проблемы многокритериального выбора коммерческой платформы системы дистанционного обучения (СДО), предполагающий построение множества недоминируемых альтернатив на основе нечеткого отношения предпочтения для поддержки проекта создания автоматизированной системы дистанционного обучения по спутниковой навигации «ГЛОНАСС-КВАЛИФИКАЦИЯ».

Постановку задачи в краткой форме представим следующим образом. Пусть задано множество альтернатив A и каждая альтернатива характеризуется несколькими критериями качества с номерами $j = 1, \dots, m$. Информация о попарном сравнении альтернатив по каждому критерию качества j представлена в форме отношения предпочтения R_j . Таким образом, имеется m отношений предпочтения R_j на множестве A . Требуется выбрать лучшую альтернативу из множества $\{A, R_1, \dots, R_m\}$.

Для формализации метода многокритериального выбора альтернатив на основе нечеткого отношения предпочтения укажем ряд определений.

Определение 1. Нечетким отношением R на множестве A называется нечеткое подмножество декартова произведения $A \times A$, характеризующееся функцией принадлежности $\mu_R : A \times A \rightarrow [0,1]$. Значение $\mu_R(a, b)$ этой функции понимается как степень выполнения отношения $a \wedge b$.

Определение 2. Нечетким отношением предпочтения на A называется любое заданное на этом множестве рефлексивное нечеткое отношение, функция принадлежности которого вычисляется следующим образом:

$$\mu_{R^s}(a, b) = \begin{cases} \mu_R(a, b) - \mu_R(b, a), & \text{если } \mu_R(a, b) \geq \mu_R(b, a); \\ 0, & \text{если } \mu_R(a, b) < \mu_R(b, a). \end{cases}$$

Определение 3. Пусть A — множество альтернатив и μ_R — заданное на нем нечеткое отношение предпочтения. Нечеткое подмножество недоминируемых альтернатив множества (A, μ_R) описывается функцией принадлежности

$$\mu_R^{nd} = 1 - \sup_{a, b \in A} (\mu_{R^s}(b, a) - \mu_{R^s}(a, b)), \quad a \in A.$$

Определение 4. Четко недоминируемыми называются альтернативы, для которых $\mu_R^{nd}(a) = 1$, а множество таких альтернатив

$$A^{chnd} = \{a | a \in A, \mu_R^{nd}(a) = 1\}.$$

Определение 5. Носителем нечеткого множества B с функцией принадлежности $\mu_B(a)$ является множество $\{a | a \in A, \mu_B(a) > 0\}$.

С учетом введенных формализмов процедура решения задачи многокритериального выбора выполняется в несколько шагов.

Шаг 1. Строится нечеткое отношение Q_1 , которое является пересечением исходных отношений предпочтения:

$$\mu_{Q_1}(a, b) = \min(\mu_{R_1}(a, b), \dots, \mu_{R_m}(a, b))$$

и определяется нечеткое подмножество недоминируемых альтернатив в множестве (A, μ_{Q_1}) :

$$\mu_{Q_1}^{nd}(a) = 1 - \sup_{b \in A} (\mu_{Q_1}(b, a) - \mu_{Q_1}(a, b)).$$

Шаг 2. Строится нечеткое отношение Q_2 :

$$\mu_{Q_2}(a, b) = \sum_{j=1}^m w_j \mu_{R_j}(a, b)$$

и определяется нечеткое подмножество недоминируемых альтернатив в множестве (A, μ_{Q_2}) :

$$\mu_{Q_2}^{nd}(a, b) = 1 - \sup_{b \in A} (\mu_{Q_2}(b, a) - \mu_{Q_2}(a, b)).$$

Данная функция упорядочивает альтернативы по степени их недоминируемости. Числа w_j в приведенной выше свертке представляют собой коэффициенты относительной важности рассматриваемых критериев, для которых выполняются следующие условия:

$$\sum_{j=1}^m w_j = 1, \quad w_j \geq 0, \quad j = \overline{1, m}.$$

Шаг 3. Отыскивается пересечение множеств $\mu_{Q_1}^{nd}$ и $\mu_{Q_2}^{nd}$:

$$\mu^{nd}(a) = \min(\mu_{Q_1}^{nd}(a), \mu_{Q_2}^{nd}(a)).$$

Шаг 4. Рациональным считается выбор альтернатив из множества

$$A^{nd} = \{a | a' \in A, \mu^{nd}(a') = \sup_{a \in A} \mu^{nd}(a)\}.$$

Решение: Наиболее рациональной альтернативой из множества A^{nd} является та, которая имеет максимальную степень недоминируемости.

2. Решение задачи многокритериального выбора

Задачу выбора наиболее предпочтительной платформы СДО для поддержки проекта «ГЛОНАСС-КВАЛИФИКАЦИЯ» с учетом наиболее важных критериев качества рассмотрим на примере анализа широко представленных на коммерческом рынке следующих альтернативных корпоративных систем дистанционного обучения:

Система дистанционного обучения *TrainingWare*

(<http://www.ksob.ru>)

СДО *TrainingWare* предназначена для создания единой среды обучения, взаимодействия обучаемых с преподавателем и между собой, организации проведения различных видов обучения, контроля компетенции, проведения регулярных тренингов и аттестаций, автоматизации входного тестирования.

Система ориентирована на предприятия, имеющие разветвленную структуру и имеющие потребность в организации дистанционного обучения, системы повышения квалификации, удаленного обучения, а также тестирования и сертификации.

Система *TrainingWare* - это современное программное обеспечение, позволяющее развернуть в Интернет- или Интранет-сетях учебные центры дистанционного обучения. Созданные с помощью *TrainingWare* учебные центры обеспечивают весь цикл дистанционного обучения, включая:

- регистрацию учебных курсов, слушателей и инструкторов, ведение их личных дел;
 - публикацию учебных материалов в различной форме, создание и публикацию упражнений и тестов;
 - учет успеваемости в форме электронной ведомости или электронной карточки сотрудника соответственно;
 - формирование и ведение синхронизованного по времени учебного процесса и расписания;
 - другие необходимые администраторские и пользовательские сервисы и свойства.
 - организацию процесса дистанционного обучения во всех необходимых аспектах;
 - оперативное обновление информационного наполнения учебных курсов и программ;
 - планирование процесса обучения, в зависимости от индивидуальной степени подготовки и потребностей в знаниях;
 - обучение индивидуально или в группах;
 - отслеживание процессов самостоятельной подготовки, автоматический контроль времени потребляемого сотрудником на самообучение и тестирование;
 - интерактивное обучение и тренинги;
 - создание развитой системы тестирования, с учетом индивидуальных характеристик.
- Технологические достоинства TrainingWare:
- Система внедряется в открытом коде (PHP).
 - Система устанавливается под наиболее распространенные типы ОС, включая семейство Unix, Linux.
 - Система масштабируется при использовании таких СУБД, как MSSQL, Oracle.
 - Система не требует покупки дополнительного программного обеспечения или дополнительных лицензий (Windows, MS IIS и т.п.).
 - Наличие программы качества ISO 9000.

Система дистанционного обучения WebTutor

(<http://www.websoft.ru/db/wb/AFA6AE6928A1EBB1C3256C0D002A69D9/doc.html>)

Система дистанционного обучения WebTutor состоит из следующих модулей:

- 1) Модуль управления дистанционным обучением:
 - редактор учебных курсов;

- редактор интерактивных упражнений;
 - редактор тестов/контрольных вопросов;
- 2) Модуль управления учебным порталом:
 - редактор информационных материалов портала;
 - хранилище организационной структуры/ведение пользователей;
 - управление/модерирование форумов;
 - 3) Шлюз для обмена с корпоративными данными:
 - загрузка данных из системы учета персонала;
 - интеграция с (Active Directory, Dimino Directory, LDAP);
 - экспорт данных в хранилище данных, построенное на основе любой реляционной базы данных.

Поддерживает международные стандарты обмена учебными материалами (SCORM, AICC), заявлена возможность построения на основе ПО WebTutor распределенной системы дистанционного обучения для компаний с филиальной сетью любого масштаба.

Система дистанционного обучения "Прометей"

(<http://www.prometeus.ru>)

Система "Прометей" имеет модульную архитектуру, что позволяет расширять, модернизировать и масштабировать систему по мере необходимости. Система состоит из следующих модулей.

- Типовой Web-узел - набор HTML-страниц, предоставляющих информацию об учебном центре, списке курсов и дисциплин, списке тьюторов в Интернете или ЛВС (Интранете) организации.
- АРМ "Администратор". Модуль обеспечивает выполнение администратором своих служебных обязанностей. К обязанностям относятся: управление системой, разграничение прав доступа к ее компонентам, регистрация новых тьюторов и организаторов. Пользователь может работать с любого клиентского компьютера, подключенного к Сети.
- АРМ "Организатор". Модуль обеспечивает выполнение организатором своих служебных обязанностей. К обязанностям относятся: формирование групп учащихся, регистрация слушателей, контроль за оплатой обучения и рассылкой учебных материалов. Пользователь может работать с любого клиентского компьютера, подключенного к Сети.
- АРМ "Тьютор". Модуль обеспечивает выполнение тьютором своих служебных обязанностей. К обязанностям относятся: консультирование слушателей, контроль за их успеваемостью, тестирование, простановка оценок в зачетную книжку, формирование

отчетов руководству. Пользователь может работать с любого клиентского компьютера, подключенного к Сети.

- АРМ "Слушатель". Модуль обеспечивает слушателя всеми необходимыми средствами для успешного изучения курса. Слушатель может общаться с тьютором и однокурсниками, изучать электронные версии курсов, выполнять лабораторные работы, сдавать тесты, работать над ошибками. Пользователь может работать с любого клиентского компьютера, подключенного к Сети.

- Модуль "Трекинг". Модуль фиксирует в базе данных все обращения к информационным материалам, расположенным на Web-сервере учебного центра, предоставляя отчетность о том кто, когда и что читал или просматривал.

- Модуль "Курс". Модуль обеспечивает доступ к курсам со стороны слушателей, тьюторов, организаторов и администратора. Для каждого пользователя список курсов формируется динамически на основании его членства в группах.

- Модуль "Регистрация". Модуль регистрирует в системе новых слушателей и вносит информацию о них в базу данных.

- Модуль "Тест". Модуль формирует для каждого слушателя уникальное тестовое задание. Сохраняет ответы на вопросы в базе данных, анализирует их и подсчитывает набранный балл. Генерирует подробный отчет о попытке сдачи теста и сохраняет его на сервере для последующего анализа.

- Модуль "Дизайнер тестов". Модуль позволяет в интерактивном режиме создавать новые тесты, расширять и изменять существующие или импортировать тест из текстового файла. Пользователь может работать с любого клиентского компьютера, подключенного к Сети.

- Модуль "Учет". Модуль обеспечивает контроль за поступлением платежей и рассылкой учебных материалов.

- Модуль "Отчеты". Модуль формирует разнообразные отчеты о деятельности учебного заведения.

- Модуль "Дизайнер курсов". Модуль позволяет в автономном режиме создавать электронные учебные курсы с их последующим размещением на сервере учебного центра. Представляет собой отдельную программу, устанавливаемую на локальный компьютер. Подключение этого компьютера к Сети не обязательно.

Система дистанционного тренинга RedClass

(<http://www.redcenter.ru/?sid=336>)

Система дистанционного тренинга REDCLASS версии - это комплекс программно-аппаратных средств, учебных материалов и мето-

дик обучения, которые позволяют дистанционно обучаться, повышать квалификацию, контролировать знания в любых отраслях деятельности человека, а также вырабатывать практические навыки по эксплуатации и управлению программными продуктами, оборудованием и технологиями.

Для решения задач дистанционного тренинга REDCLASS обладает следующими средствами:

- Среда эмуляции упражнений позволяет формировать и проверять навыки работы слушателей с системами, обладающими оконным интерфейсом. Например, может эмулировать работу MS Word для обучения пользователей этой системы. Упражнения для среды эмуляции создаются в Конструкторе упражнений. Конструктор упражнений позволяет создавать упражнения с разветвленным сценарием выполнения и различными системами оценивания действий пользователей.

- Виртуальные лаборатории предоставляют слушателям возможность работы с реальными (не эмулированными) программно-аппаратными комплексами (стендами) в удаленном режиме. Оборудование находится в учебном центре, а слушатели получают доступ к нему со своего рабочего места. Виртуальные лаборатории позволяют дать практику самостоятельной работы, не ограниченной возможностями эмулятора.

- Электронный учебник предназначен для доставки мультимедийного контента на рабочее место слушателя. Разработка курсов ведется в автономной среде, именуемой Авторской системой.

- Система тестирования предназначена для контроля успеваемости слушателей. В системе предусмотрены средства тестирования, позволяющие осуществлять входной, выходной и промежуточный контроль знаний, а также самооценку.

- Система управления процессом обучения позволяет организовать процесс обучения в части управления каталогом курсов, пользователями Системы и их правами доступа, отчетностью, системными каталогами (режимов обучения, внешних ресурсов, методик создания курса и т.п.).

СДТ REDCLASS сертифицирована по международному стандарту SCORM 1.2**.

Для оценки альтернатив используем восемь критериев качества:

F₁ — стоимость;

F₂ — функциональность;

F₃ — надежность;

F₄ — удобство использования;

F₅ — качество технической поддержки;
 F₆ — модульность;
 F₇ — масштабируемость;
 F₈ — кросс-платформенность.

На основании функций принадлежности всех альтернатив по восьми критериям определены экспертным способом их конкретные нормированные значения, которые представляют собой следующие нечеткие множества:

$$\begin{aligned} \mu_{F_1} &= 0,25/1,5 + 0,25/1,5 + 0,1/1,8 + 0/2,2; \\ \mu_{F_2} &= 1,0/10 + 1,0/15 + 1,0/15 + 0,5/5,0; \\ \mu_{F_3} &= 0,9/0,2 + 1,0/0,1 + 0,1/0,5 + 0/0,7; \\ \mu_{F_4} &= 0,9/1,5 + 0,25/3,5 + 0,25/3,5 + 0,75/2,5; \\ \mu_{F_5} &= 0,15/2,0 + 1,0/6,0 + 1,0/6,0 + 0,75/4,5; \\ \mu_{F_6} &= 0,25/18 + 0,2/20 + 0,8/12 + 0,9/10; \\ \mu_{F_7} &= 0,95/1,0 + 0,25/7,0 + 1,0/9,0 + 0,75/5,0; \\ \mu_{F_8} &= 0,75/6,0 + 0,5/3,0 + 0,5/3,0 + 0,5/3,0. \end{aligned}$$

По этим данным составлены матрицы нечетких отношений предпочтения R₁, ..., R₈.

| | | | |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| $\mu_{R_1} = a_2$ | $\begin{matrix} F_1 & a_1 & a_2 & a_3 & a_4 \\ a_1 & 1 & 0 & 0,15 & 0,25 \\ a_2 & 0 & 1 & 0,15 & 0,25 \\ a_3 & 0 & 0 & 1 & 0,1 \\ a_4 & 0 & 0 & 0 & 1 \end{matrix}$ | $\mu_{R_2} = a_2$ | $\begin{matrix} F_2 & a_1 & a_2 & a_3 & a_4 \\ a_1 & 1 & 0 & 0 & 0,5 \\ a_2 & 0 & 1 & 0 & 0,5 \\ a_3 & 0 & 0 & 1 & 0,5 \\ a_4 & 0 & 0 & 0 & 1 \end{matrix}$ |
| $\mu_{R_3} = a_2$ | $\begin{matrix} F_3 & a_1 & a_2 & a_3 & a_4 \\ a_1 & 1 & 0 & 0,8 & 0,9 \\ a_2 & 0,1 & 1 & 0,9 & 1,0 \\ a_3 & 0 & 0 & 1 & 0,1 \\ a_4 & 0 & 0 & 0 & 1 \end{matrix}$ | $\mu_{R_4} = a_2$ | $\begin{matrix} F_4 & a_1 & a_2 & a_3 & a_4 \\ a_1 & 1 & 0,65 & 0,65 & 0,15 \\ a_2 & 0 & 1 & 0 & 0 \\ a_3 & 0 & 0 & 1 & 0 \\ a_4 & 0 & 0,5 & 0,5 & 1 \end{matrix}$ |

| | | | |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| $\mu_{R_5} = a_2$ | $\begin{matrix} F_5 & a_1 & a_2 & a_3 & a_4 \\ a_1 & 1 & 0 & 0 & 0 \\ a_2 & 0,85 & 1 & 0 & 0,25 \\ a_3 & 0,85 & 0 & 1 & 0,25 \\ a_4 & 0,6 & 0 & 0 & 1 \end{matrix}$ | $\mu_{R_6} = a_2$ | $\begin{matrix} F_6 & a_1 & a_2 & a_3 & a_4 \\ a_1 & 1 & 0,05 & 0 & 0 \\ a_2 & 0 & 1 & 0 & 0 \\ a_3 & 0,5 & 0,55 & 1 & 0 \\ a_4 & 0,65 & 0,7 & 0,15 & 1 \end{matrix}$ |
| $\mu_{R_7} = a_2$ | $\begin{matrix} F_7 & a_1 & a_2 & a_3 & a_4 \\ a_1 & 1 & 0,7 & 0,85 & 0,2 \\ a_2 & 0 & 1 & 0,15 & 0 \\ a_3 & 0 & 0 & 1 & 0 \\ a_4 & 0 & 0,5 & 0,65 & 1 \end{matrix}$ | $\mu_{R_8} = a_2$ | $\begin{matrix} F_8 & a_1 & a_2 & a_3 & a_4 \\ a_1 & 1 & 0,25 & 0,25 & 0,25 \\ a_2 & 0 & 1 & 0 & 0 \\ a_3 & 0 & 0 & 1 & 0 \\ a_4 & 0 & 0 & 0 & 1 \end{matrix}$ |

Задача выбора решаем в соответствии с описанной выше процедурой нечеткого отношения предпочтения.

Шаг 1. Строим нечеткое отношение Q₁=R₁ ∩ R₂ ∩ ... ∩ R₈:

$$\mu_{Q_1}(a_i, a_j) = \begin{matrix} a_1 & a_2 & a_3 & a_4 \\ a_1 & 1 & 0 & 0 & 0 \\ a_2 & 0 & 1 & 0 & 0 \\ a_3 & 0 & 0 & 1 & 0 \\ a_4 & 0 & 0 & 0 & 1 \end{matrix}.$$

Находим подмножество недоминируемых альтернатив на множестве {A, μ_{Q₁}}:

$$\begin{aligned} \mu_{Q_1}(a_i) &= 1 - \sup_{a_j \in A} (\mu_{Q_1}(a_j, a_i) - \mu_{Q_1}(a_i, a_j)) \text{ по всем } i \text{ и } j (i \neq j); \\ \mu_{Q_1}^{HD}(a_1) &= 1 - \sup(\mu_{Q_1}(a_2, a_1) - \mu_{Q_1}(a_1, a_2); \mu_{Q_1}(a_3, a_1) - \mu_{Q_1}(a_1, a_3); \\ \mu_{Q_1}(a_4, a_1) - \mu_{Q_1}(a_1, a_4)) &= 1; \\ \mu_{Q_1}^{HD}(a_2) &= 1 - \sup(\mu_{Q_1}(a_1, a_2) - \mu_{Q_1}(a_2, a_1); \mu_{Q_1}(a_3, a_2) - \mu_{Q_1}(a_2, a_3); \\ \mu_{Q_1}(a_4, a_2) - \mu_{Q_1}(a_2, a_4)) &= 1; \\ \mu_{Q_1}^{HD}(a_3) &= 1 - \sup(\mu_{Q_1}(a_1, a_3) - \mu_{Q_1}(a_3, a_1); \mu_{Q_1}(a_2, a_3) - \mu_{Q_1}(a_3, a_2); \\ \mu_{Q_1}(a_4, a_3) - \mu_{Q_1}(a_3, a_4)) &= 1; \end{aligned}$$

$$\mu_{Q_1}^{нл}(a_4) = 1 - \sup(\mu_{Q_1}(a_1, a_4) - \mu_{Q_1}(a_4, a_1); \mu_{Q_1}(a_2, a_4) - \mu_{Q_1}(a_4, a_2); \mu_{Q_1}(a_3, a_4) - \mu_{Q_1}(a_4, a_3)) = 1.$$

Таким образом,

$$\mu_{Q_1}^{нл} = \|1 \ 1 \ 1 \ 1\|.$$

Шаг 2. Строим отношение Q_2 :

$$\mu_{Q_2}(a_i, a_j) = \sum_{k=1}^8 w_k \mu_{R_k}(a_i, a_j).$$

Коэффициенты w_k относительной важности критериев имеют следующие значения: $w_1 = 0,23$, $w_2 = 0,09$, $w_3 = 0,04$, $w_4 = 0,23$, $w_5 = 0,04$, $w_6 = 0,09$, $w_7 = 0,23$, $w_8 = 0,04$.

Определяем нечеткое отношение Q_2 :

$$\mu_{Q_2}(a_i, a_j) = \begin{matrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{matrix} \begin{vmatrix} 1 & 0,325 & 0,427 & 0,229 \\ 0,038 & 1 & 0,105 & 0,152 \\ 0,079 & 0,05 & 1 & 0,082 \\ 0,082 & 0,293 & 0,278 & 1 \end{vmatrix}.$$

Находим подмножество недоминируемых альтернатив множества $\{A, \mu_{Q_2}\}$:

$$\mu_{Q_2}^{нл}(a_i) = 1 - \sup_{a_j \in A} (\mu_{Q_2}(a_j, a_i) - \mu_{Q_2}(a_i, a_j)) \text{ по всем } i \text{ и } j (i \neq j);$$

$$\mu_{Q_2}^{нл}(a_i) = \|1,0 \ 0,712 \ 0,652 \ 0,854\|.$$

Шаг 3. Результирующее множество недоминируемых альтернатив есть пересечение множеств $\mu_{Q_1}^{нл}$ и $\mu_{Q_2}^{нл}$:

$$\mu_{Q_1}^{нл} \cap \mu_{Q_2}^{нл} = \{(1 \ 1 \ 1 \ 1) \cap (1,0 \ 0,712 \ 0,652 \ 0,854)\} = \{(1,0 \ 0,712 \ 0,652 \ 0,854)\}.$$

Шаг 4. Следовательно, рациональным следует считать выбор альтернативы a_1 , имеющей максимальную степень недоминируемости.

Заключение

Представленные результаты анализа и решения задачи многокритериального выбора программной платформы для построения и развития виртуальной системы дистанционного

обучения (для поддержки проекта «ГЛОНАСС-КВАЛИФИКАЦИЯ») с использованием метода нечеткого отношения предпочтения показали большее предпочтение системе дистанционного обучения TrainingWare. При этом имеется возможность уточнения полученного решения посредством решения указанной задачи по другим критериям, а также использование других нечетких методов решения задачи, например, на основе пересечения нечетких множеств, с использованием правил нечеткого вывода, на основе аддитивной свертки, с использованием метода ранжирования альтернатив на множестве лингвистических векторных оценок и др. [4,5].

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Техническое задание на ОКР «Разработка автоматизированной системы дистанционного обучения подготовки и повышения квалификации специалистов по спутниковой навигации». М.: ОАО «Российские космические системы», 2009. - 43 с.
2. Соловов А.В. Электронное обучение: проблематика, дидактика, технология. – Самара: «Новая техника», 2006. – 454 с.
3. Новый подход к инженерному образованию: теория и практика открытого доступа к распределенным информационным и техническим ресурсам. Под редакцией А.А.Полякова. М.: Центр-Пресс, 2000. – 238 с.
4. Андрейчинков А.В., Андрейчинкова О.Н. Анализ, синтез, планирование решений в экономике. М. «Финансы и статистика», 2000. - 368 с.
5. Теория и практика неопределенного программирования. Б.Лю; пер. с англ. – М.: БИНОМ. Лаборатория знаний, 2009. – 416 с.

В.П. КОРЯЧКО, Р.А. ТАГАНОВ, С.В. ЧЕРНЫШЕВ

Рязанский государственный радиотехнический университет

МЕТОДОЛОГИЧЕСКИЕ ПРИНЦИПЫ ПОСТРОЕНИЯ ВИРТУАЛЬНЫХ УЧЕБНЫХ ЛАБОРАТОРИЙ

Рассматриваются методологические принципы построения виртуальных учебных лабораторий, которые ориентированы на формирование практических умений и развитие профессионально-ориентированной интуиции в процессе обучения.

Введение

Поддержка лабораторного практикума традиционно является важной составляющей проблематики электронного обучения. Так, раз-

рабатываемые в 80-е годы учебно-исследовательские САПР и АСНИ, наряду с курсовым и дипломным проектированием, широко применялись и в учебных лабораториях. В последние годы эти вопросы вновь начинают широко обсуждаться, но уже в более четкой, конкретной формулировке термина «Виртуальные учебные лаборатории» (ВУЛ) [1-4].

Применительно к профессиональному образованию концепция ВУЛ в потенциале ориентирована на реализацию требований к компьютеризации профессиональной подготовки, соответствует идеям открытого и дистанционного обучения и позволяет, хотя бы частично, сгладить остроту существующих ныне проблем материально-технического обеспечения учебного процесса.

1. Методологические подходы и принципы

Немногочисленные пока научно-методические работы по тематике ВУЛ ограничены, в основном, описанием виртуальных приборов и лабораторных занятий с их использованием. Но понятие ВУЛ для профессионального образования гораздо шире и может интегрировать в себя не только виртуальные приборы, но и виртуальные учебные кабинеты натуральных объектов, системы математического и имитационного моделирования, учебные и промышленные пакеты прикладных программ, компоненты CALS-систем. ВУЛ могут использоваться не только в лабораторном практикуме, но и в курсовом и дипломном проектировании, в учебно-исследовательской работе студентов.

В методологическом плане виртуальные лаборатории можно классифицировать, исходя из принятой в системах искусственного интеллекта типологии моделей представления знаний, на системы процедурного, декларативного и гибридного (процедурно-декларативного) типов.

1.1. Виртуальные лаборатории процедурного типа

Основу ВУЛ процедурного типа составляют учебные пакеты прикладных программ или их промышленные аналоги, предназначенные для автоматизации профессионального труда. При их создании основное внимание обычно обращают на реализацию алгоритмов математического моделирования, расчета и оптимизации изучаемых объектов или процессов. Порой математическое моделирование является единственным способом учебного исследования сложных объектов или процессов. Кроме того, сами по себе методы и средства профессиональной деятельности могут быть в таких ВУЛ предметом самостоятельного изучения, например, методы геометрического моделирования, алгоритмы оптимизации и т.п. Другим примером ВУЛ проце-

дурного типа может считаться лаборатория по методам системного моделирования, анализа предметной области автоматизации и проектирования программных систем на основе CALS (ИПИ)- и CASE (САПР) – технологий [5-10].

При всей несомненной полезности автоматизация профессионального труда в учебных задачах не всегда приводит к повышению качества собственно профессиональной подготовки. Помочь здесь может специальный дидактический интерфейс, сценарные схемы которого могут базироваться на принципах построения сценариев интеллектуальных тренажеров, да и сами такие тренажеры вполне могут быть отнесены к ВУЛ процедурного типа.

1.2. Виртуальные лаборатории декларативного типа

К ВУЛ декларативного типа могут быть отнесены виртуальные учебные кабинеты, поскольку знания в них хранятся в готовом, препарированном виде. Как уже отмечалось выше, эти виртуальные кабинеты сходны по подготовке и работе с ними с электронными учебниками. Но их содержательными прототипами являются не первоисточники на бумаге, а натурные экспонаты реальных учебных кабинетов, которые нередко называют учебными лабораториями [11-13].

1.3. Виртуальные лаборатории гибридного типа

Гибридный подход к построению ВУЛ применяют обычно при разработке виртуальных приборов. При этом внешняя атрибутика, в частности панель управления, отображается визуально адекватно ее реальному аналогу, а различные режимы работы исследуются с помощью математических или имитационных моделей.

На протяжении ряда лет на кафедре САПР ВС РГРТУ указанные подходы использовались при разработке виртуальных тренажеров по технологии микроэлектронного производства, а также при разработке виртуальных кабинетов и лабораторий по ИПИ- и САПР-технологиям.

Наглядным примером этого типа ВУЛ является разработанный программный комплекс виртуальных технологических устройств, имитирующих работу технологических автоматов микроэлектронного производства:

- виртуальная установка по эпитаксии кремниевых пластин;
- виртуальная установка по технологии диффузии;
- виртуальная установка по технологии тонких пленок;
- виртуальные средства по технологии неразъемных соединений.

Еще одно перспективное направление создания гибридных ВУЛ - имитация типовых лабораторных работ на сложном и уникальном оборудовании, например, на электронном микроскопе. Обычная си-

туация при традиционном проведении таких лабораторных работ - все манипуляции с оборудованием проводит штатный сотрудник лаборатории, преподаватель дает пояснения, а студенты наблюдают и, в лучшем случае, проводят обработку результатов экспериментов. Причем эксперимент проводится, как правило, лишь для одного набора исходных параметров, а для других вариантов студентам даются уже готовые результаты.

Следовательно, подготовив компьютерную базу экспериментальных данных для различных исходных параметров и набор видеофрагментов реальных экспериментов, совсем нетрудно разработать виртуальную интерактивную установку для проведения типовых лабораторных исследований с помощью компьютера.

1.4. Лаборатории удаленного доступа

Особое место в электронном обучении занимают лаборатории удаленного доступа, создаваемые в ряде ведущих вузов России и в том числе в РГРТУ на основе уникальных лабораторных стендов и промышленных установок. Автоматизированное лабораторное оборудование, входящее в состав таких лабораторий, может работать под управлением учащихся с удаленных рабочих мест, подключенных к компьютерной сети. Реализация такого подхода позволяет организовать самостоятельную работу учащихся при выполнении индивидуальных заданий или их одновременную работу не только с учебно-лабораторным оборудованием, но и с научно-исследовательскими установками, например, работу на электронном микроскопе или работу на электронном оборудовании системы ГЛОНАСС [4].

Заключение

Таким образом, виртуальную лабораторию можно рассматривать как аппаратно-программный инструментальный используемый в качестве объектно-ориентированной информационной среды для эффективного интерактивного взаимодействия пользователя со средой моделирования. Это позволяет обучающемуся, расположенному на любом расстоянии от объекта, в интерактивном режиме оперативно конструировать в операционной среде компьютера изображение передней панели нужного инструмента или группы инструментов – испытательную лабораторию для выполнения требуемой задачи.

Важным преимуществом виртуальной лаборатории является возможность наглядной имитации реального физического эксперимента путем использования, наряду с обычными изображениями приборов еще и полученных ранее реальных экспериментальных данных, хранящихся в базе данных виртуальной лаборатории.

Наиболее рациональным и эффективным в дидактическом плане представляется комплексный подход к созданию ВУЛ, который предлагается к использованию для проекта по разработке автоматизированной системы дистанционного обучения подготовки и повышения квалификации специалистов по спутниковой навигации [4], где требуется и обеспечивается поддержка обучения на всех этапах познавательного процесса.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Соловов А.В. Электронное обучение: проблематика, дидактика, технология. – Самара: «Новая техника», 2006. – 454 с.
2. Новый подход к инженерному образованию: теория и практика открытого доступа к распределенным информационным и техническим ресурсам. Под редакцией А.А.Полякова. М.: Центр-Пресс, 2000. – 238 с.
3. Корячко В.П., Моисеенко В.П., Таганов А.И. Информационные технологии подготовки специалистов в области ИПИ (CALS) – технологий // Актуальные проблемы ракетно-космического приборостроения и информационных технологий. Материалы Всероссийской НТК. Москва: РНИИ КП, 2008.
4. Техническое задание на ОКР «Разработка автоматизированной системы дистанционного обучения подготовки и повышения квалификации специалистов по спутниковой навигации». М.: ОАО «Российские космические системы», 2009.
5. Корячко В. П., Таганов А.И., Таганов Р.А. Информационно-образовательная среда системного моделирования и управления проектами а CALS (ИОС-CALS)/ Свидетельство об официальной регистрации программы для ЭВМ №2002610045 (Россия). Зарегистрировано в РосАПО 16.01.2002, заявка № 2001611589.
6. Корячко В.П., Таганов А.И., Таганов Р.А. База знаний ВУЗа: методы процессного моделирования для проектов внедрения ИПИ технологий // Заявка на отраслевую регистрацию разработки в ОФАП № 7096 от 26.10.2006.
7. Корячко В.П., Таганов А.И., Таганов Р.А. База знаний ВУЗа: методы информационного моделирования для проектов внедрения ИПИ технологий // Заявка на отраслевую регистрацию разработки в ОФАП № 7095 от 26.10.2006.
8. Корячко В.П., Таганов А.И., Таганов Р.А. База знаний ВУЗа: методы онтологического моделирования для проектов внедрения ИПИ технологий // Заявка на отраслевую регистрацию разработки в ОФАП № 7093 от 26.10.2006.

9. Корячко В.П., Таганов А.И., Таганов Р.А. База знаний ВУЗа: методы объектно-ориентированного моделирования для проектов внедрения ИПИ технологий// Заявка на отраслевую регистрацию разработки в ОФАП № 7150 от 26.10.2006.

10. Корячко В.П., Таганов А.И., Таганов Р.А. База знаний ВУЗа: методы обследования и анализа объекта автоматизации для проектов внедрения ИПИ технологий// Заявка на отраслевую регистрацию разработки в ОФАП № 7151 от 26.10.2006.

11. Корячко В.П., Таганов А.И., Таганов Р.А. База знаний ВУЗа: методы сбора информации для проектов внедрения ИПИ технологий// Заявка на отраслевую регистрацию разработки в ОФАП № 7152 от 26.10.2006.

12. Корячко В.П., Таганов А.И., Таганов Р.А. База знаний ВУЗа: методы функционального моделирования для проектов внедрения ИПИ технологий// Заявка на отраслевую регистрацию разработки в ОФАП № 7146 от 26.10.2006.

13. Корячко В.П., Таганов А.И., Таганов Р.А. База знаний ВУЗа: регламенты применения инструментальных средств управления данными об изделии в проектах внедрения ИПИ - технологий// Заявка на отраслевую регистрацию разработки в ОФАП № 7147 от 26.10.2006.

С.В. АНИКЕЕВ, В.Л. ДУБИНИНА, А.В. МАРКИН

Рязанский государственный радиотехнический университет

О СРЕДСТВАХ ГЕНЕРАЦИИ ОТЧЕТОВ

Статья посвящена анализу возможностей современных программных средств генерации отчетов. Целью статьи является не только дать обзор наиболее распространенных средств создания отчетов, но также выделить основные достоинства и недостатки каждого из генераторов, что поможет правильно выбрать наиболее подходящее для каждого конкретного проекта средство генерации отчетов.

В настоящее время стремительно растет число пользователей, нуждающихся в информации, и с развитием компьютерных технологий они ожидают высокой производительности и доступности аналитических приложений. Разработка и реализация отчетов для приложений обычно рассматривается как одна из многих связанных с разработкой задач. Решить проблему средствами среды разработки информационной системы (ИС) крайне сложно, и в таком случае используются специализированные генераторы отчетов [1].

Современный генератор отчетов – это одновременно и набор компонентов, доступных разработчику и позволяющих управлять генерацией отчетов из приложения, и отдельный модуль, способный генерировать отчеты для баз данных ИС. Современные генераторы отчетов представляют собой инструменты обработки данных, по сложности и функциональности приближающиеся к средствам разработки приложений баз данных (БД) [2].

Генератор отчетов – особый класс программ, востребованный как программистами, так и пользователями – всеми, кому необходим подробный отчет по данным в нужное время и в требуемой форме [3]. Каждое из средств генерации отчетов отличается своим способом дизайна, разработки и поставки отчетов с приложением. Генераторы отчетов существуют как в виде генераторов, встроенных в системы управления реляционными базами данных (СУБД) или среды разработки, так и в виде самостоятельных программ.

Встроенные генераторы отчетов

Общей особенностью всех встроенных средств генерации отчетности является то, что они могут использоваться для создания отчетов только в той среде разработки, в которую интегрированы.

Создание отчетов в “настольных” СУБД. Visual FoxPro и MS Access являются СУБД для работы с локальными БД. В этих СУБД предусмотрены встроенные средства создания отчетов [4, 5].

Для создания простейших отчетов с автоматически располагаемыми полями в Visual FoxPro используется *стандартный отчет (Quick Report)*, а в MS Access – *Автоотчет*. Также для создания отчетов может использоваться *мастер отчетов (Report Wizard)*, позволяющий быстро создать не сложный отчет, выбрав поля для отчета, параметры сортировки и группировки данных, стиль отображения данных и их расположения. Для построения сложных отчетов предназначен *конструктор отчетов*, с помощью которого можно провести предварительную сортировку и группировку данных, форматирование данных, подсчет промежуточных результатов, а затем просмотреть результат в окне предварительного просмотра.

Таким образом, можно говорить об удобстве и простоте создания отчетов с помощью средств рассматриваемых СУБД. Но, конечно, возможности создания сложных аналитических отчетов в таких СУБД невелики. Помимо этого такие генераторы поддерживают небольшое количество источников данных и форматов экспорта. Можно сделать вывод, что формирование отчетности средствами Visual FoxPro и MS Access может использоваться только для простых задач.

Генератор отчетов Rave Reports в Delphi. В качестве основного средства создания отчетов и их использования в приложениях в состав Delphi включен генератор отчетов Rave Reports.

В состав Rave Reports входят ядро генератора отчетов, визуальная среда создания отчетов и набор компонентов [6]. *Ядро генератора отчетов* обеспечивает управление отчетом, его предварительный просмотр, печать. Исполняемый код ядра включается в файл приложения Delphi, делая его полностью автономным при работе с отчетами на компьютере клиента. *Визуальная среда* позволяет создавать самые разнообразные отчеты, в том числе использующие наборы данных из источников различных типов. Отчеты сохраняются в файлах с расширением gav и должны распространяться совместно с приложениями, использующими их. *Набор компонентов* предоставляет разработчику инструментарий для управления отчетом в приложении.

Безусловно, средства генератора Rave Reports заметно упрощают процесс создания отчетов и позволяют разработать достаточно сложные отчеты. Однако Rave Reports – это встроенный генератор отчетов, и, соответственно, он может использоваться только для разработки отчетов для приложений Delphi.

Создание отчетов в системе 1С:Предприятие. Система 1С:Предприятие широко используется для автоматизации различных областей деятельности предприятий – управленческого учета, бухгалтерского учета, учета движения средств и т.д.

Система 1С: Предприятие содержит различные средства для получения обобщенных данных и создания различного рода отчетов. Отличительной особенностью является то, что совокупность этих средств представляет собой определенную иерархию, состоящую из нескольких уровней [7]. Причем часть средств может быть использована не только разработчиком, но и конечным пользователем системы.

Самым нижним уровнем иерархии средств для построения отчетов является *уровень встроенного языка* 1С:Предприятие. На этом уровне основной инструмент для получения информации из БД – язык запросов. Наряду с этим используются объекты, позволяющие разработчику организовывать интерактивное взаимодействие с пользователем в процессе формирования запроса и вывода полученных данных. К таким «интеллектуальным» объектам относится, например, *построитель отчета*. Основой его работы является текст запроса, который перед выполнением может быть изменен пользователем в режиме 1С:Предприятия (пользователь может настроить запрос «под себя»).

Следующим уровнем средств для построения отчетов является *уровень визуального конструирования*. На этом уровне используются

два инструмента: *конструктор запроса* и *конструктор выходной формы*. Конструктор запроса позволяет составлять и редактировать текст запроса (который уже имеется в программе), исключительно визуальными средствами, не требуя знания синтаксиса языка запросов. Конструктор выходной формы является еще более универсальным инструментом. Он использует конструктор запроса и позволяет создать полностью готовый отчет: форму, макет, набор процедур.

Третий, самый верхний, уровень иерархии – *интерактивный уровень*. На этом уровне в режиме 1С:Предприятия используются две внешние обработки: *консоль запросов*, предназначенная для составления запросов и анализа полученных результатов, и *консоль отчетов*, позволяющая создавать и оформлять отчеты любой сложности. Использование этих обработок позволяет создавать отчеты, не изменяя конфигурацию, прямо в режиме 1С:Предприятия.

Таким образом, возможность по созданию сложных отчетов в системе 1С:Предприятие достаточно большая, и предоставляются средства для упрощения разработки отчетов. Однако так как речь идет о встроенном в приложение генераторе отчетов, то он может подключаться только к своему источнику данных (базе данных 1С).

Службы Reporting Services в MS SQL Server. Несмотря на то, что рынок «наводнен» различными инструментальными средствами отчетности, компания Microsoft создала собственный инструмент отчетности – службы Reporting Services [8].

Построение отчета представляет собой процесс, состоящий из создания соединения, определения возвращаемых данных и последующего упорядочивания данных. После построения отчет можно развернуть на Web-сервере отчетности, где к нему могут получить доступ другие пользователи. Обращение к отчетам возможно через HTTP из Internet Explorer или из приложения как к Web-службе.

Службы Reporting Services встроены в MS SQL Server, однако предназначены не только для данных на основе MS SQL Server. Источниками данных для отчетов могут быть также Excel, Access, Oracle, плоские файлы или любые источники баз данных OLE или Open Database Connectivity (ODBC, открытый интерфейс взаимодействия с базами данных). Имеется возможность встраивать отчеты в приложения.

Службы Reporting Services постоянно улучшаются, их функциональность расширяется, но, тем не менее, отстает от функциональности универсальных инструментальных средств. Данные службы спроектированы и настроены все-таки именно для MS SQL Server. Также следует отметить, что Reporting Services очень ресурсоемкий продукт, требует установленной платформы .NET на сервере и SQL Server.

Формирование отчетов средствами Oracle. Oracle – это современная СУБД, позволяющая создавать приложения практически любой сложности и содержащая функционально полный набор интегрированных средств для проектирования БД, разработки аналитических приложений, разработки отчетов и для решения других задач.

Oracle Reports – это гибкое визуальное средство разработки отчетов, которые могут иметь весьма сложную структуру, содержать результаты нескольких запросов, автоматически формировать итоги и подитоги, включать в себя разнообразную графическую и ссылочную информацию. Как правило, отчеты готовятся квалифицированными специалистами и затем выполняются конечными пользователями.

Масштабируемая многоуровневая архитектура Oracle Reports позволяет оптимизировать загрузку, устанавливая сервер отчетов Reports Server на мощном центральном компьютере и освобождая ресурсы клиентских рабочих станций [9]. Отчеты также могут формироваться при возникновении заранее определенного события, могут планироваться для выполнения в периоды низкой загрузки.

Reports позволяет получать данные из текстовых файлов и различных реляционных СУБД через стандартные интерфейсы (ODBC и JDBC). Oracle Reports позволяет легко распространять информацию во всей организации, используя Web-архитектуру. Важно отметить, что Oracle Reports не требует от разработчиков применения нескольких подходов для проектирования отчетов, выполняемых в различных средах. Единожды созданный отчет будет выполняться на большинстве платформ, а его выходная форма представлена в разных форматах – от текстовых файлов до динамических Web-страниц.

Таким образом, Oracle Reports – мощное средство, однако "заточенное" под Oracle и дорогое. Oracle Reports тесно интегрирован с Oracle Database, использует одинаковые с ним языки разработки. Также следует отметить, что занимает много места при установке.

Oracle Discoverer – это инструмент для получения произвольных отчетов, формирования нерегламентированных запросов и анализа данных. Он обеспечивает быстрый и удобный доступ к информации, содержащейся в реляционных и многомерных хранилищах данных, а также в транзакционных системах, в том числе не обязательно работающих под управлением СУБД Oracle [9].

В отличие от Oracle Reports, где конечный пользователь является только потребителем информации, в Oracle Discoverer он может самостоятельно получать необходимые данные в требуемых аналитических разрезах. Для повышения производительности в Discoverer реализован целый ряд технологических возможностей: предсказание времени вы-

полнения запроса до его начала, возможность создания и последующего автоматического использования суммарных таблиц, интеллектуальный механизм кэширования. Эти технологии обеспечивают приемлемые времена отклика даже при работе с большими объемами данных.

Таким образом, Oracle Discoverer обеспечивает уникальное сочетание простоты использования, администрирования и производительности. Его применение быстро и при небольших затратах дает ощутимый результат – после простого внедрения организация сразу получает выгоду от упрощившегося доступа пользователей к информации.

Независимые генераторы отчетов

Независимые генераторы отчетов непосредственно не связаны с какими-либо средами разработки, СУБД. Данные продукты являются отдельными, и в них включается дизайнер отчетов и небольшой макроязык. Они имеют гораздо больше возможностей, но для использования требуют наличия ряда библиотек DLL [10].

Генератор отчетов Crystal Reports. Среди генераторов отчетов – самостоятельных программ самым мощным, несомненно, является Crystal Reports фирмы Seagate.

Crystal Reports является универсальной программой, способной удовлетворить запросы различных пользователей. Он имеет мощный инструментарий для создания сложных отчетов, позволяющий выполнять очень сложную статистическую обработку и объединять данные из разнородных источников, предоставляет мощный визуальный конструктор отчетов, средства распространения созданных отчетов, средства встраивания отчетов в приложения, средства публикации в Internet информации БД.

Очень важным отличием между Crystal Reports и, например, собственными средствами создания отчетов Delphi является метод взаимодействия Crystal с данными [10]. Большинство интегрированных средств создания отчетов используют прямую связь с файлами данных, из-за чего процесс создания отчета может неблагоприятно повлиять на визуальные компоненты средств управления, использующие тот же файл данных. Crystal создает внутреннюю, хранящуюся в оперативной памяти таблицу, состоящую из выбранных для отчета таблиц и полей, и затем к данной таблице применяются все необходимые критерии сортировки, фильтрации и т.д. Так как данный процесс является внутренним для Crystal и не требует участия пользователя, он значительно ускоряет вывод данных.

К *недостаткам* можно отнести большое количество и большие размеры файлов, необходимых при распространении отчетов. Также следует отметить, что Crystal Reports приобретается отдельно от

Delphi/C++Builder, при этом новые версии продукта и компонент часто несовместимы со старыми версиями.

Формирование отчетов средствами OracleBI Publisher. Среди самостоятельных продуктов компании Oracle следует отметить Oracle Business Intelligence Suite Enterprise Edition, объединяющий несколько продуктов, которые могут использоваться как вместе, так и независимо друг от друга [11]. Среди этих продуктов для формирования отчетов предназначен OracleBI Publisher – хорошо масштабируемый сервер формирования отчетов, позволяющий генерировать отчеты в разных форматах на основе данных из множества источников и рассылать их по различным каналам. OracleBI Publisher помогает сократить высокие расходы, связанные с разработкой и сопровождением отчетов различных форматов, одновременно повышая эффективность управления отчетами. Кроме этого, он позволяет уменьшить зависимость компании от программных систем других производителей, которые требуются для форматирования документов. Oracle Publisher обеспечивает разработчиков средствами точного контроля за форматом, разметкой и выводом отчетов, позволяя создавать и распространять отчеты полиграфического качества, независимо от уровня графической сложности. Пользователи могут быстро и легко создавать отчеты с помощью OracleBI Publisher, устанавливать формат данных для отчета, планировать задачи по формированию отчетов, указывать способ вывода и канал доставки отчета, публиковать отчеты в репозитории документов.

OracleBI Publisher по возможностям можно сравнить с Crystal Reports, но использование в комплекте с другими инструментами Oracle BI позволяет использовать это средство более интенсивно и с меньшими трудозатратами на создание отчетов, чем Crystal Reports.

Генератор отчетов FastReport. FastReport, также как и Crystal Reports, является независимым генератором отчетов (отдельной программой). При этом сами отчеты встраиваются в приложение, и на пользовательских компьютерах не требуется инсталляции генератора отчетов для обеспечения работы с отчетами.

FastReport – это полностью визуальный генератор отчетов, который предоставляет множество средств для создания сложных отчетов.

Основной концепцией FastReport является высокая скорость получения отчетности. Для этого реализованы высокие скорости обработки данных (программная составляющая), обучения работе с продуктом (интерфейсная составляющая), внедрения (простота работы и понимания разработчиками программного обеспечения, документация для разработчиков), конструирования отчетов (интуитивность опера-

ций), получения конечного результата (встроенные возможности конвертации, множество форматов для экспорта и доставки отчетов) [12].

Таким образом, можно выделить следующие основные положительные характеристики FastReport [13]:

- *мощность и гибкость*, обеспечиваемые широким выбором средств разработки отчетов,

- *компактность и производительность*, обеспечиваемые рекордно высокой скоростью формирования отчетов, достаточно малым размером (FastReport добавляет к приложению меньше Кб, чем любой другой генератор отчетов со сравнимой функциональностью),

- *доступность исходного кода*,

- *подробная документация* на русском языке.

На сегодняшний день FastReport стал фактическим стандартом создания отчетности как в узкоспециализированных областях, так для корпораций малого и среднего бизнеса. Эффективность, функциональность и простота работы – главные характеристики FastReport.

Заключение

В настоящее время программы генерации отчетов являются высоко востребованными. Разными производителями предлагается множество различных инструментальных средств отчетности, каждое из которых характеризуется определенными возможностями. Однако, несмотря на обилие предлагаемых решений, проблема эффективной разработки отчетов не становится менее актуальной. Некоторые генераторы просты и удобны в использовании, однако не предоставляют достаточно возможностей по разработке сложных аналитических отчетов, другие – позволяют разработать сложнейшие отчеты, но требуют значительных программных и аппаратных ресурсов, достаточно глубоких знаний языка разработки и т.д. Помимо этого следует помнить, что даже в генераторах, предоставляющих огромный инструментарий для разработки отчетов, мало внимания уделяется первичной обработке и подготовке данных для отчетов. Таким образом, выбор того или иного генератора отчетов в каждом конкретном случае должен быть обоснован и продиктован спецификой и сложностью решаемых задач, а также объемом ресурсов и средств, которые могут быть затрачены на реализацию подсистемы отчетности. Необходимо учитывать как возможности систем, так и стоимость внедрения и поддержки решений.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Маклаков С.В., Матвеев Д.В. Анализ данных. Генератор отчетов Crystal Reports. – СПб.: БХВ-Петербург, 2003. – 496 с.

2. А. Боровский. Русский ответ Криссталлу: http://soft.mail.ru/article_page.php?id=148.
3. Обзор FastReport - генератора отчетов: <http://www.ixbt.com/soft/fastreport.shtml>.
4. Каратыгин С.А., Тихонов А.Ф., Тихонова Л.Н. Visual FoxPro 7. К вершинам мастерства. – М.: Бином-Пресс, 2003. – 648 с.
5. Горев А., Ахаян Р., Макашарипов С. Эффективная работа с СУБД. – СПб.: Питер Ком, 2006. – 704 с.
6. Генератор отчетов Rave Reports: <http://256bit.ru/Delphi7/Glava23/Index2.html>.
7. Габец А.П., Гончаров Д.И., Козырев Д.В., Кухлевский Д.С., Радченко М.Г. Профессиональная разработка в системе 1С: Предприятие 8 / Под ред. М.Г. Радченко. – М.: «1С-Публишинг»; СПб.: Питер, 2007. – 808 с.
8. Дьюсон Р. SQL Server 2008 для начинающих разработчиков: Пер. с англ. – СПб.: БХВ-Петербург, 2009. – 704 с.
9. Системы поддержки принятия решений Oracle: <http://www.olap.ru/desc/oracle/odss.asp>.
10. Обзор генераторов отчетов: анализ от TPrinter до Crystal Reports: <http://www.interface.ru/home.asp?artId=4215>.
11. Oracle Business Intelligence: http://www.caseclub.ru/zip/oracle_bi_suite_enterprise.pdf
12. Fast Reports: http://www.itpedia.ru/index.php/Fast_Reports.
13. <http://fast-report.com/ru/products/report-generator-for-delphi-fastreport-4.html>.

С.В. АНИКЕЕВ, В.Л. ДУБИНИНА, А.В. МАРКИН

Рязанский государственный радиотехнический университет

О ТЕХНОЛОГИЯХ СОЗДАНИЯ ОТЧЕТОВ В ИНФОРМАЦИОННЫХ СИСТЕМАХ

В настоящей статье рассматривается проблема удобного и быстрого доступа к данным, необходимым для формирования отчетов в информационных системах. Предлагаются различные подходы к извлечению и предварительной обработке данных для отчетов.

Одной из основных функций современных информационных систем (ИС) является создание отчетов. Однако структура современных реляционных систем управления базами данных (СУБД), на которых основаны ИС, ориентирована в первую очередь на компактное и

непротиворечивое хранение информации, а не на оптимизацию произвольной выборки и представления данных [1]. Как бы эффективно не было организовано хранение данных, одного этого не достаточно для обеспечения решения проблемы реализации удобного и быстрого извлечения требуемой информации.

Общеизвестным решением проблемы быстрой обработки сложных запросов к базе данных является использование технологии OLAP (*Online Analytical Processing*, аналитическая обработка в реальном времени). OLAP-структура, содержащая в себе базовые данные и данные об измерениях (агрегатах), потенциально предоставляет всю информацию, которая может потребоваться для ответов на любые запросы. Заявленное время обработки запросов в OLAP составляет около 0.1 % от аналогичных запросов в реляционную базу данных (БД) [2]. Решения для аналитической обработки данных имеются у фирм-разработчиков крупных коммерческих СУБД (Microsoft Analysis Services, Oracle OLAP Option и т.д.).

Однако существуют определенные сложности в применении OLAP. Одна из них – в создании запросов, выборе базовых данных и разработке схемы, в результате чего большинство современных продуктов OLAP поставляются вместе с огромным количеством предварительно настроенных запросов. Другая проблема – в базовых данных, которые обязательно должны быть полными и непротиворечивыми.

Таким образом, часто оказывается нецелесообразным использовать существующие аналитические сервисы. В большинстве случаев огромное значение приобретают алгоритмы процедур доступа к хранимым данным. Часто самым тривиальным решением является разработка запроса на выборку данных для отчета, который напрямую обращается к базовым таблицам БД. Однако на практике в большинстве случаев такой подход к извлечению данных неэффективен, так как сложные запросы, выполняемые на клиенте, приводят к низкой производительности, а кроме того иногда достаточно трудно получить данные в требуемом виде с помощью SQL-запроса. Таким образом, данные, непосредственно извлекаемые из реляционных таблиц и предоставляемые для формирования отчетов, должны быть предварительно обработаны. Для такой предварительной обработки данных могут использоваться хранимые процедуры (ХП) или представления.

Разработка отчетов с использованием представлений

Представление (view) – это виртуальная таблица, содержимое которой динамически вычисляется на основании запроса к реальным (базовым) таблицам БД. Изменение данных в базовых таблицах отра-

жается в содержимом всех представлений, построенных на основании этих таблиц [3].

Применение рассматриваемого подхода предполагает создать совокупность представлений, каждое из которых возвращает некоторое множество логически взаимосвязанных данных. Получение данных для отчетов при таком подходе выполняется не с помощью сложных запросов к базовым таблицам, а посредством простейших запросов из заранее созданных представлений. Представления позволяют скрыть сложность запросов и саму структуру таблиц БД [3]. На основе информации, полученной через представления, осуществляется разработка новых отчетных форм.

Использование представлений обеспечивает решение ряда задач, а именно – обеспечение безопасного доступа к объектам БД (фильтрация информации, доступной пользователям, предотвращение непосредственного обращения к таблицам), предоставление упрощенного и настраиваемого доступа к данным, повышение быстродействия (благодаря оптимизации и предварительной компиляции SQL-запроса, выбирающего данные представления) [3, 4].

Однако подход с использованием представлений не лишен *недостатков*. Среди них такие, как отсутствие возможности передачи в представления входных параметров, отсутствие возможности создания индексированных представлений (в настоящее время для большинства реализаций СУБД), для вложенных представлений – сложность управления, увеличение продолжительности выполнения и необходимость дополнительного места на диске для их материализации.

При неудачном проектировании структуры механизма получения информации на основе представлений, в БД может содержаться значительное их количество. Поэтому необходимо внимательно изучить имеющиеся запросы и найти в них повторяющиеся подзапросы или выражения, они и будут хорошими кандидатами в представления.

Разработка отчетов на хранимых процедурах

Хранимой процедурой (stored procedure) называется компилируемая программа произвольной длины на процедурном языке СУБД, которая хранится в БД как часть метаданных. ХП позволяют вести поиск и обработку данных непосредственно на сервере, обеспечивая максимальную независимость клиентской части приложений. Использование ХП имеет ряд преимуществ: возможность вести итеративную обработку информации, повышение производительности, локализация изменений, простота доступа, улучшенная безопасность и т.д. [5].

Рассмотрим несколько различных технологий создания отчетов на ХП. Следующие технологии внедрены и успешно применяются при разработке отчетов в информационной системе «Абонент» [6]:

- разработка для каждого отчета отдельной хранимой процедуры, реализующей всю бизнес-логику обработки извлекаемых данных;
- разработка управляемых хранимых процедур, каждая из которых извлекает и обрабатывает определенную совокупность логически взаимосвязанных данных и затем служит источником необходимых данных для процедур, разрабатываемых для конкретного отчета;
- разработка нескольких обобщенных хранимых процедур, каждая из которых извлекает и обрабатывает данные по некоторой достаточно большой группе логически взаимосвязанной информации и затем помещает эти данные во временные таблицы.

Помимо этого затронем возможности создания отчетов на динамических хранимых процедурах, реализуемых с помощью конструкции EXECUTE BLOCK.

Разработка процедур "под отчет". При использовании такой технологии предполагается, что для каждого отчета разрабатывается отдельная хранимая процедура выбора необходимых данных, реализующая всю логику их обработки.

Как показывает практика, такой подход к обработке данных обладает существенными *недостатками*. К ним относятся:

- многократное *дублирование кода хранимых процедур* (вследствие повторной разработки уже ранее реализованных алгоритмов для каждого нового отчета), что служит источником ошибок при модификации процедур;
- *достаточная сложность сопровождения* из-за чрезмерного количества ХП в БД и большого числа уровней вложенности ХП;
- *отсутствие возможности создания отчетов пользователями*.

Вместе с тем рассмотренная технология имеет некоторые преимущества. Даже сложные отчеты, использующие для выборки данных SQL-запросы с множеством соединений (join), выполняются достаточно быстро за счет того, что вся логика обработки данных сосредоточена в ХП (т.е. выполняется на сервере), а не в самом запросе, с помощью которого формируется набор данных, или кода отчета.

Разработка отчетов на управляемых хранимых процедурах. Данная технология предполагает, что в БД созданы одна или несколько (в зависимости от того, насколько широкую предметную область охватывает база данных) управляемых ХП, каждая из которых извлекает и обрабатывает по определенному алгоритму некоторую совокупность логически взаимосвязанных данных. При этом эти ХП имеют

входные параметры, с помощью которых можно управлять объемом информации, извлекаемым процедурой из БД, а также логикой необходимой обработки данных по заданным алгоритмам. Управляемые ХП являются процедурами выбора. Кроме этого для каждого отчета, также как и при первом рассмотренном подходе, разрабатывается своя ХП выбора, которая, однако, не реализует никакой логики обработки извлекаемой информации. Такая процедура задает значения управляющих параметров для управляемых ХП и выводит все поля, необходимые для построения отчета. Таким образом, данные, передающиеся в ХП, предназначенную для конкретного отчета, уже являются предварительно обработанными. Благодаря этому при разработке каждого отчета нет необходимости заботиться о программной реализации алгоритмов получения необходимых данных, нужно только выбрать необходимую информацию из управляемых процедур.

Условно можно выделить два следующих этапа получения информации для отчета:

- 1) выборка из БД логически взаимосвязанных данных управляемыми ХП;
- 2) выборка нужных полей процедурами, разрабатываемыми для конкретного отчета.

Фактически, алгоритм разработки нового отчета в рамках рассматриваемого подхода сводится к следующему.

1. Анализируется задача на создание каждого нового отчета и разрабатывается ХП, которая выбирает данные с помощью управляемых ХП. Такие процедуры являются достаточно короткими и не сложными для построения. Предполагается, что код управляемых ХП разработан заранее и изменяется редко. Таким образом, не стоит задача реализации каких-то сложных алгоритмов вычисления данных, необходимо лишь выбрать информацию из БД с помощью управляемых процедур.

2. Создается SQL-запрос на выборку из БД с помощью ХП, созданной на предыдущем этапе алгоритма.

Таким образом, можно выделить следующие основные *преимущества* данного подхода по сравнению с предыдущим:

- *исключение дублирования кода хранимых процедур* за счет реализации алгоритмов обработки каждой единицы информации только в одной из хранимых процедур (как следствие упрощается и решение проблемы сопровождения такого механизма);
- *решается проблема поиска необходимых для извлечения данных пользователями.*

Существенным *недостатком* такого подхода является значительное снижение скорости извлечения информации. Это происходит из-за часто возникающей потребности объединения в одном запросе большого количества ХП. Этот недостаток не позволяет использовать такой подход в ИС, специфика работы которых ориентирована на вывод в отчетах большого количества разнообразной информации.

Разработка отчетов на обобщенных хранимых процедурах. Этот подход предполагает, что в результате предварительного анализа предметной области выявляются наиболее общие группы логически взаимосвязанной информации. Затем на их основе в БД создаются несколько основных ХП (обобщенные ХП), которые возвращают всю необходимую информацию по каждой выявленной группе в виде временных денормализованных таблиц. Следует отметить, что в отличие от описанных выше подходов, в рамках рассматриваемой технологии создаются не процедуры выбора, а выполняемые ХП, которые записывают данные во временные таблицы. Использование временных таблиц имеет определенные преимущества. В такие таблицы можно загрузить результаты выполнения сложного длительного запроса и затем работать с результатами на протяжении всего сеанса, что существенно повышает производительность. При этом система, работая с данными временных таблиц, не блокирует от других пользователей базовые таблицы, на которых построен запрос.

При использовании данного подхода не стоит задача разработки отдельных ХП для каждого нового отчета (как в предыдущих случаях). Обобщенные ХП создаются заранее, после анализа предметной области, и меняются редко. Для получения данных необходимо последовательно выполнить два действия.

1. Сначала нужно вызвать необходимые обобщенные ХП с помощью оператора EXECUTE PROCEDURE. В результате выполнения этих процедур во временные таблицы будут записаны данные. Вызов обобщенных ХП выполняется перед формированием набора данных для отчета.

2. Затем следует разработать SQL-запрос, который выберет необходимые данные из временных таблиц.

Основными *преимуществами* данной технологии являются исключение дублирования исходного кода, отсутствие вложенности ХП, возможность создания отчетов силами самих пользователей.

Недостатком такого подхода является некоторое снижение скорости извлечения данных из-за накладных расходов, связанных с переносом данных во временные таблицы. Отдельно следует отметить особенности модификации обобщенных ХП. С одной стороны такие

ХП достаточно тяжело модифицировать из-за объединения в одной ХП большого количества элементов обработки данных, с другой стороны модификация обобщенных ХП, по сравнению с другими подходами, является наименее трудоемкой, так как логика выборки данных не дублируется и изменения чаще всего достаточно проводить только в одном месте.

Разработка отчетов на динамических хранимых процедурах.

В практической разработке программ для работы с БД часто возникает необходимость в отображении данных ИС, основанных на выполнении перекрестных запросов (Cross-Tab) к объектам базы данных. Перекрестный отчет (сводный отчет, кросс-отчет) – это отчет, который имеет табличную структуру, причем заранее неизвестно, сколько строк и столбцов будет содержать таблица отчета [7].

Основной проблемой решения Cross-Tab задач часто является то, что логическим решением выступает набор данных с переменным числом столбцов. Сформировать тривиальный запрос SELECT не составляет труда, но реализовать сложную бизнес-логику можно только с помощью ХП, которые, однако, не могут иметь переменное число столбцов. Выходом из ситуации может быть "статическое" написание столбцов в ХП, при этом недостаток такого подхода – жестко зашитые в ХП данные и логика "разворота" кросс-отчета, в результате чего может возникать необходимость переделывать некоторые отчеты.

Решением проблемы могут выступать так называемые динамические ХП или ХП без имени, описываемые конструкцией языка SQL EXECUTE BLOCK, которая доступна в некоторых СУБД (например, в СУБД Firebird 2.1) [8]. Данный оператор может использоваться в клиентских приложениях для реализации требуемой логики работы системы, когда в используемой БД нет необходимой именованной ХП. В операторе EXECUTE BLOCK доступны для использования все те же конструкции (операторы, запросы), что и в обычной ХП. Входные параметры передаются в оператор EXECUTE BLOCK из приложения, а выходные – возвращаются клиентскому приложению.

Таким образом, использование динамических ХП является одним из решений проблемы разработки сложных кросс-отчетов.

Заключение

Проблема эффективного доступа к данным, на основе которых строятся отчеты в ИС, чрезвычайно актуальна. В настоящей статье в рамках поиска решения данной проблемы предложены различные технологии извлечения и обработки хранимых данных, которые могут быть использованы как основа разработки отчетов в ИС. Каждый из описанных подходов оптимизирует разработку и формирование отчетов

тов по сравнению с написанием запросов, извлекающих данные непосредственно из базовых таблиц БД. По результатам дальнейших исследований планируется классифицировать отчеты определенным образом и затем выработать рекомендации по применению того или иного подхода для разработки отчетов определенного вида.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Маклаков С.В. Создание информационных систем с AllFusion Modeling Suite. - М.: ИНФРА-М, 2007. - 217 с.
2. OLAP: <http://ru.wikipedia.org/wiki/OLAP>.
3. Представление (базы данных): <http://ru.wikipedia.org/wiki/View>.
4. Анна Иванова. Безопасность в Microsoft SQL Server 2005: http://zeus.sai.msu.ru:7000/database/mssql/mssql_2005_security/part3.shtml
5. Борри Х. Firebird: руководство разработчика баз данных: пер. с англ. – СПб.: БХВ-Петербург, 2006. – 1104 с.
6. <http://www.abonentplus.ru>.
7. Перекрестные (cross-tab) отчеты: <http://fast-report.com/ru/documentation/frx3user/3/3-1.htm>.
8. Firebird 2.0 на полную катушку: <http://www.delphiplus.org/articles/ib/firebird-2-full-capability/1/index.html>.

А.В. БАКУЛЕВ, А.А. МИТРОШИН

Рязанский государственный радиотехнический университет

О ВОЗМОЖНОСТИ ИСПОЛЬЗОВАНИЯ СУЩЕСТВУЮЩИХ РЕСУРСОВ INTERNET ПРИ СОЗДАНИИ WEB-ПОРТАЛА СИСТЕМЫ ДИСТАНЦИОННОГО ОБУЧЕНИЯ ГЛОНАСС

Рассматриваются вопросы, связанные с использованием информационных ресурсов, размещенных в русскоязычном Internet, как дополнительных учебных материалов при изучении использования глобальной навигационной спутниковой системы ГЛОНАСС.

При создании системы дистанционного обучения, посвященной изучению возможностей использования ГЛОНАСС, определенное значение приобретает возможность использования для образовательных целей информации, размещаемой на сайтах организаций, так или иначе связанных с ГЛОНАСС. Размещаемая на таких сайтах информация, безусловно, не может использоваться непосредственно в учебных курсах, но является ценным дополнительным образовательным ресурсом. Ссылки на рассматриваемые далее источники может содержать портал

системы дистанционного обучения, посвященной изучению возможностей использования ГЛОНАСС.

Официальный сайт Прикладного потребительского Центра (ППЦ) на базе Информационно-аналитического центра (ИАЦ) ЦНИИ-маш доступен по адресу <http://www.glonass-ianc.rsa.ru>. Основной задачей этого сайта является информационная поддержка потребителей по вопросам спутниковой навигации на базе систем ГЛОНАСС (РФ) и GPS (США). Сайт поддерживает предоставление информации посетителям на русском и английском языках.

Главная страница сайта содержит новостную ленту, ссылки на разделы «ЦУС ГЛОНАСС», «ГЛОНАСС», «GPS», «Архив», «О ППЦ», ссылки на документы. Доступна также карта сайта и раздел, содержащий ссылки на адреса сайтов министерств, ведомств и организаций, являющихся разработчиками и потребителями в области спутниковой навигации. Ссылки группированы по разделам «Министерства и ведомства», «НИИ, НПО, ОКБ», «Информационные центры», «Организации и фирмы», «Международное сотрудничество». В разделе «ЦУС ГЛОНАСС» публикуются компьютерный бюллетень ЦУС «ГЛОНАСС».

Раздел «ГЛОНАСС» содержит подразделы «Состояние ОГ», «Эфемериды», «Мониторинг», «Суточный мониторинг», «Зоны видимости», «Расчет зон видимости», «Интегральная доступность» и другие.

На рис. 1 представлено содержание подраздела «Мониторинг», посвященному мониторингу текущего состояния орбитальной группировки ГЛОНАСС.



Рис. 1. Раздел «Мониторинг» сайта <http://www.glonass-ianc.rsa.ru>

На рис. 2 приведено содержание подраздела «Интегральная доступность» сайта <http://www.glonass-ianc.rsa.ru>, демонстрирующая доступность навигации наземного потребителя по системе ГЛОНАСС.

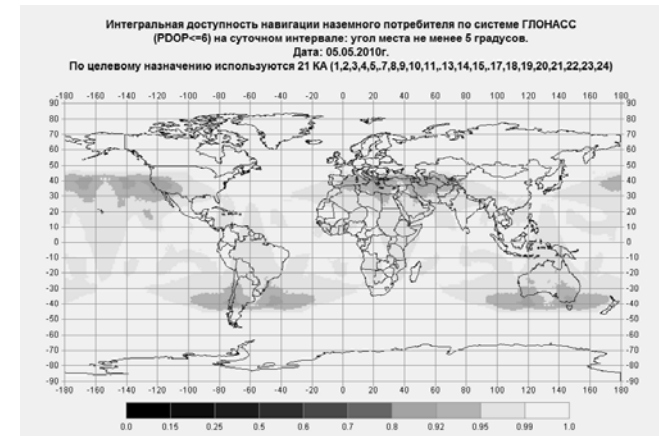


Рис. 2. Раздел «Интегральная доступность» сайта <http://www.glonass-ianc.rsa.ru>

Публикуемая информация актуальна, что делает её весьма ценной при использовании в качестве дополнительного учебного пособия при дистанционном изучении ГЛОНАСС.

В разделе «Архив» содержатся результаты анализа навигационной информации в ИАЦ, различные типы измерительной и справочной информации из международных систем сбора и хранения данных.

Сайт Ассоциации разработчиков, производителей и потребителей оборудования и приложений на основе глобальных навигационных спутниковых систем «ГЛОНАСС/ГНСС - Форум» доступен по адресу <http://www.aggf.ru>. Сайт имеет разделы «Об ассоциации», «Система ГЛОНАСС», «Законодательство», «Сертификация», «Международное сотрудничество», «Мероприятия», «Оборудование», «Интернет-журнал ИСНС», «Аналитика», «ГЛОНАСС-образование».

Раздел «Об ассоциации» содержит информацию об Ассоциации разработчиков, производителей и потребителей оборудования и приложений на основе глобальных навигационных спутниковых систем «ГЛОНАСС/ГНСС — Форум».

Раздел «Система ГЛОНАСС» содержит общую информацию о глобальной спутниковой навигационной системе, включая ссылку на «Интерфейсный контрольный документ» и библиографический спи-

сок. В разделе имеются подразделы «Историческая справка», «Документы» и «Состояние группировки».

В разделе «Законодательство» опубликованы тексты официальных документов, регламентирующих работы по ГЛОНАСС, например Распоряжение Президента № 38 от 18 февраля 1999 г. ГЛОНАСС - система двойного назначения, Федеральная целевая программа «Глобальная навигационная система» (Постановления Правительства РФ от 20.08.2001 г. №587, от 14.07.2006 г. № 423, от 12.09.2008 г. № 680) и другие.

Раздел «Сертификация» посвящен сертификации, стандартизации и лицензированию в области разработки, производства и использования оборудования и приложений на основе глобальных навигационных спутниковых систем.

В разделе «Международное сотрудничество» освещается деятельность Международного комитета по глобальным навигационным спутниковым системам. Раздел содержит подразделы «Международный комитет по ГНСС», «История создания международного комитета», «Рабочие группы международного комитета», «Основные международные мероприятия», «SIMBA II в России».

Раздел «Оборудование» содержит каталог оборудования, связанного с ГЛОНАСС. Раздел содержит подразделы «Электронные компоненты», «Базовые модули», «Антенные устройства», группы подразделов «Навигационная аппаратура потребителей» (включает подразделы «Для индивидуального использования», «Для наземного транспорта», «Для морских и речных судов», «Для железнодорожного транспорта», «Авиационная», «Космического базирования», «Геодезическая», «Аппаратура функциональных дополнений»), «Системы на основе НАП» (включает подразделы «Системы мониторинга транспорта», «Дифференциальные системы», «Системы синхронизации», «Аппаратура имитации сигналов ГНСС»), отдельный раздел позволяет систематизировать оборудование по производителям. Содержимое этого раздела может использоваться как дополнительное учебное пособие по клиентскому оборудованию ГЛОНАСС.

Раздел «Аналитика» предназначен для публикации аналитических материалов по результатам внедрения спутниковых навигационных технологий в различные сферы экономики Российской Федерации.

Раздел «ГЛОНАСС-образование» имеет непосредственное отношение к проблематике дистанционного обучения ГЛОНАСС. Содержимое раздела приведено на рис. 3.



Рис. 3. Раздел «ГЛОНАСС-образование» сайта <http://www.aggf.ru>

Сайт конструкторского бюро навигационных систем «Навис» доступен по адресу <http://www.navis.ru>. Основное назначение сайта — знакомство потенциальных потребителей с изделиями конструкторского бюро. Для целей дистанционного обучения наибольший интерес представляет раздел «Каталог». Важно отметить, что содержимое подразделов этого раздела практически полностью совпадает с содержимым подразделов раздела «Оборудование» сайта Ассоциации разработчиков, производителей и потребителей оборудования и приложений на основе глобальных навигационных спутниковых систем «ГЛОНАСС/ГНСС — Форум».

Сайт, доступный по адресу <http://www.glonass-rus.ru>, принадлежит аудиторско-консалтинговой фирме «Интегрированный бизнес сервис». Основное назначение сайта — Интернет-магазин навигационных средств. Очевидно, что информация, размещаемая в Интернет-магазине, не может использоваться для организации учебного процесса. Однако она может служить делу продвижения использования ГЛОНАСС, поэтому ссылка на этот сайт окажется полезной на портале системы дистанционного обучения ГЛОНАСС.

Таким образом, русскоязычный Интернет содержит ряд ресурсов, ссылки на которые могут быть полезны лицам, проходящим дистанционное обучение использованию спутниковой навигационной системы ГЛОНАСС. Ссылки на эти ресурсы целесообразно использовать не в

учебных курсах, а создать для них некоторый раздел в составе портала системы дистанционного обучения ГЛОНАСС.

О.М. БАСКАКОВА

Рязанский государственный радиотехнический университет

ИСПОЛЬЗОВАНИЕ СОВРЕМЕННЫХ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ ДИСТАНЦИОННОГО ОБУЧЕНИЯ НА ПРИМЕРЕ СИСТЕМЫ MOODLE

В статье рассматриваются возможности обучающей среды Moodle, а также процесс ее использования в системе дистанционного обучения Рязанского государственного радиотехнического университета.

Актуальность внедрения системы дистанционного обучения в РГРТУ определяются необходимостью соответствия образовательной деятельности университета мировым тенденциям; использования возможностей единого образовательного пространства; развития новых сегментов образовательных услуг; создания и использования новых форм обучения при их интеграции с другими формами обучения.

Для решения этих задач специалистами РГРТУ был проведен анализ предлагаемых на рынке систем дистанционного образования, из многообразия которых был выбран Moodle (Modular Object-Oriented Dynamic Learning Environment) - модульная объектно-ориентированная динамическая обучающая среда. Moodle относится к классу LMS (Learning Management System) — систем управления обучением. В нашей стране подобное программное обеспечение относят к системам дистанционного обучения (СДО). Moodle является свободно распространяемым программным обеспечением по лицензии GPL с открытым кодом, что предоставляет возможность университету бесплатно использовать систему, а также адаптировать ее в соответствии с требованиями вуза и интегрировать с другими программными продуктами.

СДО Moodle изначально ориентирована на работу преподавателей, не обладающих профессиональными знаниями в области программирования и администрирования баз данных, веб-сайтов и т.п. Поскольку Moodle имеет удобный интуитивно понятный интерфейс, он является наиболее подходящим для того, чтобы преподаватель самостоятельно, прибегая только к помощи справочной системы, смог создать электронный курс и управлять его работой. Практически во всех ресурсах и элементах курса используется удобный WYSIWYG HTML-редактор.

Moodle является web-ориентированной средой, для работы которого требуется:

- web-сервер с поддержкой PHP (например, Apache2);
- сервер баз данных (по умолчанию используется MySQL).

Рязанский государственный радиотехнический университет является техническим вузом, соответственно учебные материалы многих курсов содержат огромное количество математических формул, которые необходимо разместить в системе. Для эффективной работы преподавателей было предложено несколько способов ввода формул:

- непосредственно с помощью синтаксиса TeX;
- преобразованием в TeX формулы, созданной с помощью программы MathType;
- с помощью бесплатной программой TeXaide.

LMS Moodle позволяет вставлять таблицы, схемы, графику, видео, флэш и другие объекты. Изначально размер загружаемых файлов был ограничен до 2 Мб. С целью предотвращения возможности возникновения трудностей при попытке загрузки интерактивных элементов курса в виде файла размером, превышающим допустимый, необходимо было изменить установленное ограничение. Вносить подобные изменения имеет право только разработчик на правах системного администратора. На данный момент максимальный размер составляет 100 Мб, и при необходимости его можно изменить.

В Moodle используются следующие роли: администратор, преподаватель (разработчик курса), тьютор, студент, гость. Одной учетной записи может быть назначено несколько ролей.

В настоящее время в Moodle зарегистрированы около 40 преподавателей, для каждого из них размещены тренировочные курсы, в рамках которых отрабатываются элементы организации и проведения дистанционного обучения. Курсы распределены по категориям, соответствующим названиям кафедр. Созданы также индивидуальные учебные курсы с названиями дисциплин и каждому преподавателю в рамках авторского курса назначена роль «Разработчик курса», которая дает ему право управлять курсом, изменять и проводить обучение.

Помимо этого, в Moodle размещен курс «Типовая структура дистанционного курса», который преподаватели используют в качестве ресурса для организации обучения с применением дистанционных образовательных технологий, изучают правила работы с ними.

В перспективе планируется апробация разработанных курсов дистанционного обучения в РГРТУ и набор экспериментальных групп студентов для обучения с использованием ДО, согласование и утверждение специальностей, дисциплин, учебных программ и графиков

обучения для групп студентов, обучаемых с использованием дистанционных технологий.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Андреев А.В. Практика электронного обучения с использованием Moodle / А.В. Андреев, С.В. Андреева, И.Б. Доценко. – Таганрог: ТТИ ЮФУ, 2008. – 146 с.
2. Анисимов А.М. Работа в системе дистанционного обучения Moodle. Учебное пособие 2-е изд. испр. и дополн. / А.М. Анисимов. – Харьков, ХНАГХ, 2009. – 292 с.

А.М. ГОСТИН, Н.В. САМОХИНА, А.А. ФОКИНА

Рязанский государственный радиотехнический университет

ПРИНЦИПЫ ПОСТРОЕНИЯ СИСТЕМЫ ДИСТАНЦИОННОГО ОБУЧЕНИЯ ПО ПОДГОТОВКЕ СПЕЦИАЛИСТОВ В ОБЛАСТИ ИСПОЛЬЗОВАНИЯ ГЛОНАСС

Рассматриваются основные принципы построения разрабатываемой системы дистанционного обучения по подготовке специалистов в области использования ГЛОНАСС, используемые средства и технологии дистанционного обучения.

Для подготовки специалистов в области использования ГЛОНАСС в различных отраслях народного хозяйства в целях эффективного использования материальных и технических ресурсов необходимо использовать дистанционное обучение.

Дистанционное обучение в настоящий момент является быстро развивающейся отраслью в мире. Многие крупные компании создают центры дистанционного обучения, чтобы стандартизировать, удешевить и улучшить качество подготовки персонала. Автоматизированные системы дистанционного обучения (АСДО), использующиеся в этих центрах, отражают цели и задачи образовательных проектов и, в зависимости от этого, различаются наборами функций, используемыми коммуникационными средствами, способами взаимодействия с обучаемыми, архитектурой размещения, применяемыми технологиями дистанционного обучения, составом технических и программных средств.

Состав функций систем дистанционного обучения подбирается в зависимости от многих факторов, но существует набор базовых функций, реализуемый практически всеми системами:

- 1) управление пользователями;
- 2) создание дистанционных учебных курсов;
- 3) дистанционное проведение занятий;
- 4) контроль уровня усвоения материала;
- 5) формирование отчетов по обучению.

К коммуникационным средствам, используемым в системах дистанционного обучения, относят:

1. сетевые или средства интернет-обучения - основаны на формировании виртуального образовательного пространства с помощью компьютерных сетевых технологий;

2. кейс - доминирует использование индивидуальных комплектов учебно-методических материалов (кейсов) в электронном формате в сочетании с гибкими формами взаимодействия преподавателей и слушателей, а также мультимедийной поддержкой процесса обучения;

3. спутниково-телевизионные - основаны на формировании виртуального образовательного пространства с помощью систем телевидения и спутниковых каналов передачи данных.

По способам взаимодействия с обучаемыми различают британскую (асинхронную) и американскую (синхронную) модель обучения.

Британская модель (асинхронная, индивидуальная) разработана в Британском открытом университете и исходит исторически из заочной формы обучения; слушателям предоставляются учебные материалы не одновременно со всеми, а в удобное для них время, также предоставляется система психолого-педагогической поддержки в форме индивидуальных консультаций и активных групповых занятий.

Американская модель (синхронная) восходит к очной форме образования, в которой очный контакт преподавателя со студентом заменен на видеоконференции; слушатели работают над приобретением одинаковых знаний или навыков в одно и то же время, учебно-методическое обеспечение не отличается от учебно-методического обеспечения для очной формы; качество образования обеспечивается главным образом за счет обеспечения качества телекоммуникационных каналов между базовым учреждением (студией, в которой находится лектор) и удаленными аудиториями (классами).

Существует несколько вариантов архитектуры размещения АСДО — централизованная, распределенная и децентрализованная (без центрального сервера). При проведении классификации используется понятие узел — комплекс, в котором организовано согласованное использование компонентов АСДО.

Централизованное дистанционное обучение. Используется единый центральный узел. Учебные материалы хранятся на центральном узле АСДО. Оболочкой для прохождения курсов выступает Web-браузер. Доставка учебных материалов происходит по сети. Вся информация о ходе обучения хранится на центральном узле АСДО, поэтому возможен полный контроль над ходом обучения. Возможно масштабирование за счет увеличения производительности оборудования или масштабирования на уровне предприятия. Возможна консолидация результатов обучения для анализа.

Распределенное дистанционное обучение. Используется несколько независимых узлов, включающих основной функционал АСДО. Учебные материалы хранятся в распределенных узлах ДО. Оболочкой для прохождения курсов могут выступать как Web-браузер, так и специальные средства. Доставка учебных материалов происходит по сети. Вся информация о ходе обучения хранится на сервере, поэтому возможен полный контроль над ходом обучения. Консолидация результатов требует организационных мероприятий. Масштабирование осуществляется либо на уровне одного узла системы за счет повышения производительности оборудования, либо «клонированием» одного базового варианта. Решение по развертыванию накладывает меньшие требования к пропускной способности канала между сервером и слушателями, так как слушатели учатся на ближайшем сервере.

Децентрализованное дистанционное обучение. Используется несколько взаимосвязанных узлов. Вариант предполагает либо централизованное управление узлами системы, либо контроль над функционированием дочерних узлов. Учебные материалы хранятся на распределенных узлах АСДО. Оболочкой для прохождения курсов могут выступать как Web-браузер, так и специальные средства. Доставка учебных материалов происходит по сети. Информация о ходе обучения хранится на центральном узле АСДО, возможна организация дополнительных подчиненных узлов АСДО. Возможен полный контроль над ходом обучения. Масштабирование осуществляется добавлением нового узла в систему. На уровне одного узла масштабирование возможно за счет повышения производительности оборудования. Возможна консолидация результатов обучения для анализа. Решение по развертыванию накладывает меньшие требования к пропускной способности канала между сервером и слушателями.

Основными технологиями дистанционного обучения являются:

- чат-занятия — учебные занятия, осуществляемые с использованием чат-технологий. Чат-занятия проводятся синхронно, то есть все участники имеют одновременный доступ к чату;
- веб-занятия — дистанционные уроки, конференции, семинары, лабораторные работы, практикумы и другие формы учебных занятий, проводимых с помощью средств телекоммуникаций и других возможностей Интернет; для веб-занятий используются специализированные образовательные веб-форумы — форма работы пользователей по определённой теме или проблеме с помощью записей, оставляемых на одном из сайтов с установленной на нем соответствующей программой. От чат-занятий веб-форумы отличаются возможностью более длительной (многодневной) работы и асинхронным характером взаимодействия учеников и педагогов;
- телеконференции — проводятся, как правило, на основе списков рассылки с использованием электронной почты. Также существуют формы дистанционного обучения, при котором учебные материалы высылаются почтой в регионы.

При рассмотрении систем дистанционного обучения различного масштаба выявлен ряд параметров, по которым различается их построение:

- состав функций и задач: администрирование пользователей, разработка курсов, поддержка процесса обучения и контроля знаний, формирование отчетности и т.д.;
- используемые коммуникационные средства: кейсы учебно-методических материалов, сетевые интернет-технологии, спутниково-телевизионные технологии;
- способы взаимодействия со слушателями: синхронное, асинхронное, смешанное обучение;
- архитектура размещения: централизованная, распределенная, децентрализованная;
- применяемые технологии дистанционного обучения: мультимедийные лекции, видеолекции, лабораторные практикумы, виртуальные тренажеры;
- состав технических средств;
- состав программных средств.

1. Функциональные особенности АСДО.

Разрабатываемая АСДО предназначена для решения следующих задач:

- получение исходных данных из Информационного аналитического комплекса (ИАК) для планирования процесса подготовки специалистов в области спутниковой навигации;

- планирование учебного процесса;
- ведение базы данных пользователей;
- администрирование учебного процесса;
- мониторинг процесса обучения;
- импорт учебных материалов из Комплекса научно-методического и технического обеспечения (КНТМО);
 - разработка и публикация учебных материалов;
 - ведение каталога учебных ресурсов;
 - дистанционное проведение занятий (лекций, практических и лабораторных занятий, в том числе с использованием эмуляторов работы аппаратно-программные средства спутниковых навигационных систем (СНС) и базовых учебно-методических средств (БУМС);
 - проведение занятий в режиме автономного обучения;
 - промежуточный и итоговый контроль уровня подготовки обучаемых;
 - формирование и отправка статистики для ИАК;
 - формирование отчетов по обучению;
 - идентификация пользователей, обеспечение авторизованного доступа, распределение прав доступа к информации;
 - информационное взаимодействие со слушателями в процессе обучения: поддержка форумов, сервиса электронной почты, чатов, файловый обмен;
 - полнотекстовый поиск информационных ресурсов.

а) используемые коммуникационные средства.

Необходимо разработать автоматизированную систему дистанционного обучения, осуществляющую взаимодействие со слушателем через Интернет и автоматизирующую процесс предоставления учебных материалов слушателю, доступ к БУМС, средствам программной эмуляции реального оборудования СНС и контроль обучения. Для обучения слушателей, не имеющих постоянного доступа к сети Интернет, предлагается организовать средства off-line (автономного) обучения — кейс-средства, содержащие учебные материалы и поддерживающие процесс обучения слушателей. Средства поддержки автономного обучения должны обеспечивать интеграцию результатов обучения в АСДО.

– Способы взаимодействия со слушателями.

Предлагается асинхронное взаимодействие при изучении теоретического материала и самопроверки и синхронное взаимодействие при проведении групповых занятий и лабораторных работ с использованием БУМС. При данной организации взаимодействия слушатель сможет пройти курс обучения и тестирование в удобное для себя время в рамках графика обучения, но групповые занятия и выполнение лабораторных работ будет выполнять в строго определенное время (по расписанию) во избежании накладок при доступе к БУМС.

• Архитектура размещения.

Основные достоинства и недостатки различных вариантов размещения АСДО, представлены в табл. 1.

Табл. 1. Достоинства и недостатки вариантов размещения АСДО

| Вариант архитектуры размещения АСДО | Достоинства | Недостатки |
|-----------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------|
| Централизованное дистанционное обучение | <ul style="list-style-type: none"> • Возможность контроля процесса обучения. • Возможность планирования обучения централизованным и автоматизированными средствами. • Возможность централизованной автоматической консолидации результатов обучения. • Возможность централизованного обновления версий дистанционных курсов. | Высокие требования к пропускной способности канала между сервером ДО и всеми слушателями. |

| Вариант архитектуры размещения АСДО | Достоинства | Недостатки |
|--------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Децентрализованное дистанционное обучение без центрального сервера | <ul style="list-style-type: none"> • Возможность планирования обучения централизованными автоматизированными средствами. • Снижаются требования к каналу связи между слушателями и ближайшим сервером ДО. | <ul style="list-style-type: none"> • Отсутствие возможности автоматической централизованной консолидации результатов обучения. • Требуется дополнительное время и ресурсы на развертывание АСДО в регионах. |
| Распределенное обучение с центральным сервером | <ul style="list-style-type: none"> • Возможность контроля процесса обучения. • Возможность планирования обучения централизованными автоматизированными средствами. • Необходим канал между слушателями и ближайшим сервером ДО. • Возможность централизованного обновления версий дистанционных курсов. | <ul style="list-style-type: none"> • Сложная архитектура ПО АСДО. • Требуется дополнительное время и ресурсы на развертывание АСДО в регионах. |

С учетом имеющихся требований к функционалу и технической архитектуре программного комплекса АСДО, было выбрано централизованное решение с центральным сервером, поддерживающее различные варианты доставки учебных материалов.

▪ Технологии дистанционного обучения.

На основе требований, предъявляемым к АСДО по СНС были выбраны следующие технологии:

- сетевые электронные учебные курсы с поддержкой мультимедиа -технологий;
- тесты;
- индивидуальные задания;
- виртуальные лабораторные практикумы (с использованием эмуляторов реального оборудования СНС и БУМС).

Создание автоматизированной системы дистанционного обучения с вышеперечисленными параметрами позволит обеспечить процесс подготовки и повышения квалификации специалистов в области создания и использования спутниковых навигационных систем для субъектов РФ, заинтересованных министерств и ведомств, коммерческих структур, некоммерческих организаций и массовых потребителей путем использования автоматизированных средств и методов дистанционного обучения.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Моисеева М.В., Полат Е.С., Бухаркина М.Ю., Нежурина М.И. Интернет-обучение: технологии педагогического дизайна / Под ред. кандидата педагогических наук М.В.Моисеевой. - М.: Издательский дом «Камерон», 2004. – 216 с.
2. Тихонов А.Н., Кулагин В.П., Кузнецов Ю.М., Оболяева Н.М. и др. Современные педагогические технологии интернет-обучения: Сб. статей. - М.: Московская типография №2, 2008. – 190 с.
3. Тихонов А.Н., Васильев В.Н., Гридина Е.Г. и др. Интернет-порталы: содержание и технологии: Сб. науч. ст. Вып.1. ГНИИ ИТ «Информика». - М.: Просвещение, 2003. – 720 с.

А.М. ГОСТИН, А.А. ФОКИНА, Н.В. САМОХИНА

Рязанский государственный радиотехнический университет

СТРУКТУРНО-ФУНКЦИОНАЛЬНАЯ СХЕМА АСДО

Рассматривается построение структурно-функциональной схемы автоматизированной системы дистанционного обучения.

Выбор структуры АСДО осуществлялся на основании накопленного опыта построения систем аналогичного типа и в соответствии с

предъявляемыми к системе требованиями. Обобщенная структура представлена на рис. 1.

В состав АСДО входят:

комплекс технических средств (КТС), включающий телекоммуникационные средства, ОПО, АПС защиты информации, комплект ЗИП, серверы и РМ;

специальное программное обеспечение, в состав которого входит подсистема администрирования АСДО, подсистема управления учебным процессом, подсистема отчетности и статистики и подсистема Web-портала;

информационное обеспечение, в состав которого входят БД нормативно-справочной информации и БД АСДО;

ИЭОС, в состав которых входят информационные и программные средства.

Функциональные подсистемы включают программные средства для решения функциональных задач. Состав функциональных подсистем АСДО представлен на рисунках 2-6.

В функции **подсистемы администрирования АСДО** входит управление учетными записями пользователей АСДО. Данные слушателей АСДО передаются в подсистему администрирования из подсистемы управления учебным процессом, которая обеспечивает получение из ИАК учетных данных слушателей. Администраторы, преподаватели, методисты и кураторы регистрируются вручную администратором АСДО.

Функции подсистемы включают:

ведение базы данных пользователей,
регистрация и распределение прав доступа к ресурсам и функциям АСДО **пользователей**,
управление ролями,
регистрация слушателей в АСДО.

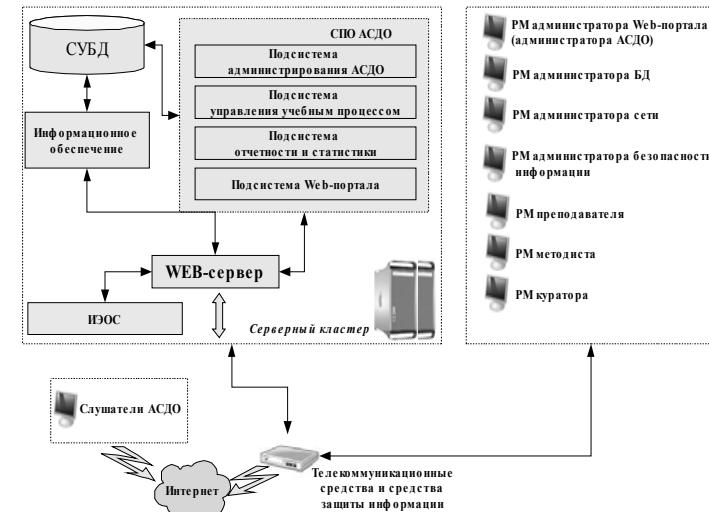


Рис. 1. Обобщенная структура АСДО

В качестве функций **подсистемы ИЭОС** можно выделить: дистанционное проведение занятий, проведение занятий в режиме автономного обучения, промежуточный и итоговый контроль уровня подготовки обучающихся,

импорт учебных материалов из КНТМО,
разработка и публикация учебных материалов,
ведение каталога учебных ресурсов.

Функции выполняются с участием методиста, преподавателя, часть функций являются автоматическими.

Подсистема Web-портала подразумевает:

авторизация, идентификация пользователей,
регламентирование прав доступа,
информационное взаимодействие со слушателями,
поиск информационных ресурсов.

Функции выполняются с участием администратора, часть функций являются автоматическими.

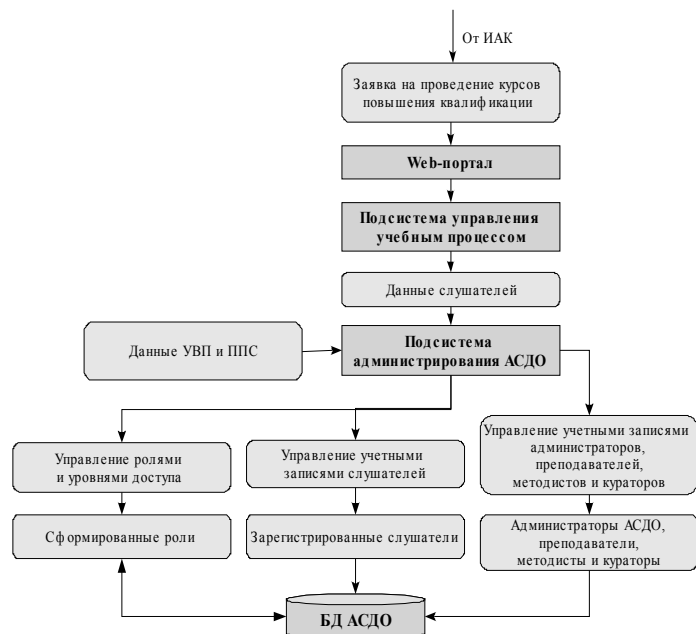


Рис. 2. Подсистема администрирования АСДО

Подсистема отчетности и статистики направлена выполнение таких функций как:

- формирование статистики для ИАК,
- формирование отчетов по обучению.

Функции выполняются с участием методиста, часть функций являются автоматическими.

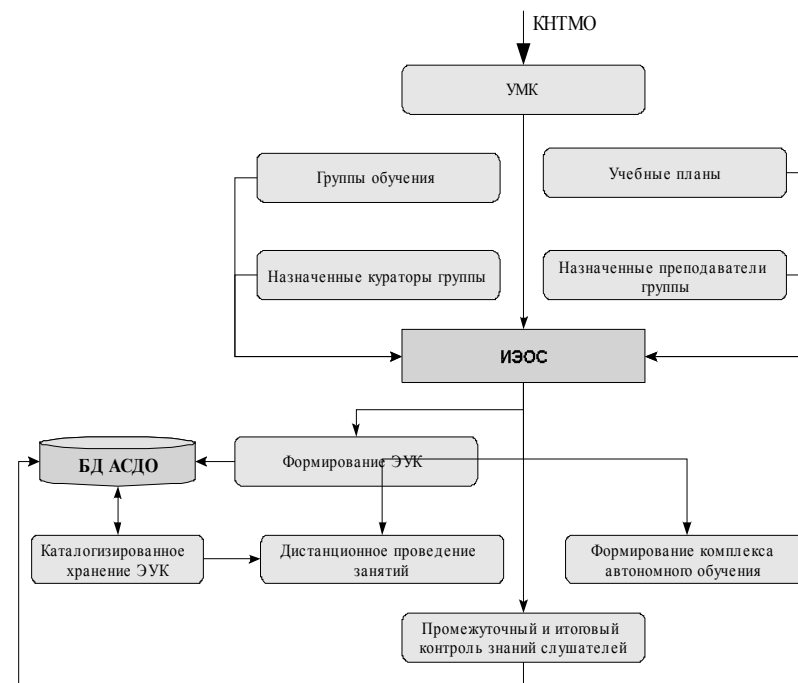


Рис.3. Интерактивные электронные обучающие системы

Функции подсистемы **управления учебным процессом** включают:

- получение исходных данных из ИАК для планирования процесса подготовки специалистов в области спутниковой навигации,
- отправка статистики для ИАК,
- планирование учебного процесса,
- администрирование учебного процесса,
- назначение и проверка индивидуальных заданий,
- проведение консультаций по вопросам курса с помощью средств коммуникации АСДО,

мониторинг процесса обучения: сбор и хранение результатов обучения, сбор и хранение данных по организационным показателям обучения, проведение текущего мониторинга пользователей.

Функции выполняются с участием методиста, куратора, преподавателя, часть функций являются автоматическими.

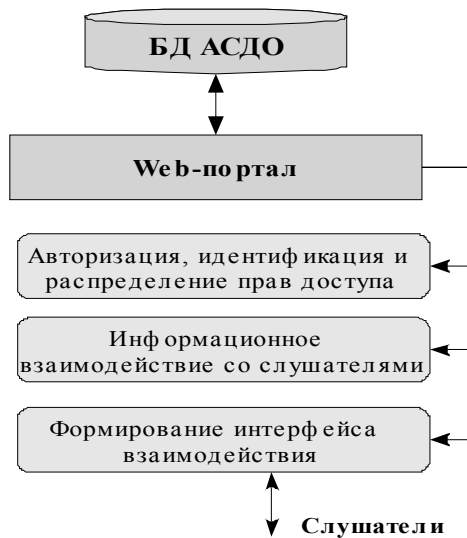


Рис. 4. Подсистема Web-портала



Рис. 5. Подсистемы отчетности и статистики

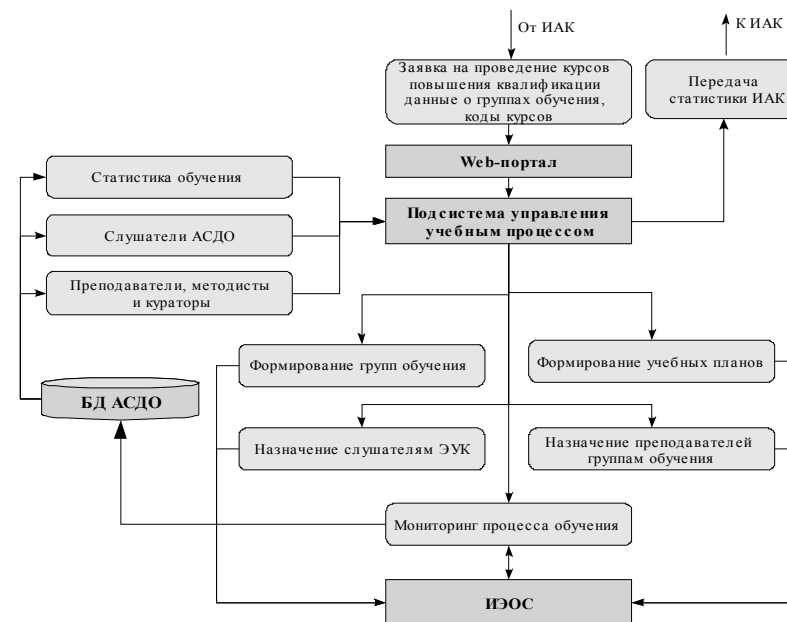


Рис. 6. Подсистема управления учебным процессом

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Моисеева М.В., Полат Е.С., Бухаркина М.Ю., Нежурина М.И. Интернет-обучение: технологии педагогического дизайна / Под ред. кандидата педагогических наук М.В.Моисеевой. - М.: Издательский дом «Камерон», 2004. – 216 с.
2. Тихонов А.Н., Кулагин В.П., Кузнецов Ю.М., Оболяева Н.М. и др. Современные педагогические технологии интернет-обучения: Сб. статей. - М.: Московская типография №2, 2008. – 190 с.
3. Тихонов А.Н., Васильев В.Н., Гридина Е.Г. и др. Интернет-порталы: содержание и технологии: Сб. науч. ст. Вып.1. ГНИИ ИТ «Информика». - М.: Просвещение, 2003. – 720 с.

А.М. ГОСТИН, Р.А. ТАГАНОВ, С.В. ЧЕРНЫШЕВ

Рязанский государственный радиотехнический университет

ПРИНЦИПЫ РАЗРАБОТКИ ВИРТУАЛЬНЫХ ИНТЕЛЛЕКТУАЛЬНЫХ ТРЕНАЖЕРОВ

Рассматриваются принципы разработки виртуальных интеллектуальных тренажеров, предназначенных для формирования практических умений и развития профессионально-ориентированной интуиции в процессе обучения в вузе.

Введение

Важной составляющей профессиональной подготовки специалистов инженерного уровня является применение в учебном процессе компьютерных систем автоматизации инженерного труда в соответствующей отрасли. Это позволяет приблизить обучаемых к реальным практическим инженерным задачам и позволяет на практике формирование необходимых практических умений, развитие профессионально-ориентированной интуиции и уверенности специалистов в своей профессиональной компетенции и возможности самостоятельно решать сложные практические задачи.

Для развития этих важных профессиональных качеств в состав современных учебных мультимедиа комплексов по дисциплинам профессиональной подготовки должны включаться наряду с традиционными средствами компьютерной поддержки процесса обучения (электронными учебниками и пакетами прикладных программ) также и программно-информационные системы, называемые интеллектуальными тренажерами [1]. Поэтому рассматриваемые здесь принципы построения таких компьютерных систем учебного назначения весьма актуальны и позволяют сформулировать на их основе содержательные этапы современной методики разработки виртуальных тренажеров.

1. Принципы построения интеллектуальных тренажеров

С учетом существующего опыта рассмотрим принципы построения особого класса тренажеров, предназначенного для формирования интеллектуальных умений, развития интуиции и творческих способностей в сфере профессиональной деятельности [1,2]. Такие тренажеры так же, как и специальные пакеты прикладных программ (САПР и др.), базируются на математических моделях изучаемых объектов и процессов. Существенной их частью является база знаний и дидактический интерфейс, позволяющий проводить интерактивную учебную работу по решению специально подобранных учебных задач в режиме детерминированного учебного исследования [1,3].

1.1. Выбор типовой учебной задачи

Первый принцип, реализуемый в построении тренажеров, - выбор типовой интересной и поучительной задачи или класса задач. Это ключевая и наименее формализуемая проблема разработки тренажера. Нет задачи - нет тренажера! Удачный выбор задачи предопределяет успех при реализации остальных принципов построения тренажера.

Учебные задачи должны соответствовать профилю учебного курса, иметь реальные прототипы и четкий физический смысл, обладать предсказуемостью результатов решения лишь в самых общих чертах. К сожалению, более четких рекомендаций дать невозможно. Внимательное изучение профессиональной предметной области позволит пробудить профессиональную интуицию, на основе которой разработчик тренажера сможет подобрать подходящую задачу или класс задач.

1.2. Обратные связи в тренажерах

Второй принцип, реализуемый в построении тренажеров – организация циклического, замкнутого управления познавательной деятельностью учащихся. Этот фундаментальный принцип общей теории управления рассмотрим применительно к тренажерам.

Напомним, что циклической, замкнутой системой управления называют систему с обратными связями. В педагогических системах образовательные системы (ОС) делят на внутренние и внешние. Информация внутренней ОС поступает к учащемуся и используется им для самокоррекции своей деятельности. Информация внешней ОС поступает к педагогу и используется им для коррекции деятельности учащегося и обучающей программы.

Понятие внутренней ОС имеет исключительно важное значение для построения тренажеров. Внутренняя ОС призвана частично заменить необходимую помощь преподавателя на этапах анализа результатов и принятия решений. Работая с тренажером, учащийся должен оперативно получать информацию о правильности (или эффективности) своих действий. Строго говоря, результаты расчетов с использованием математических моделей изучаемых объектов или процессов сами по себе являются основной информацией внутренней ОС. Но их анализ не всегда доступен учащемуся. Необходимо давать ему дополнительную информацию, которая бы стимулировала и помогала проводить вдумчивое изучение результатов расчета.

Простейшим дополнительным сообщением, которое, как показывает опыт, стимулирует интерес к анализу результатов расчета, яв-

ляется сообщение об оценке действий, выполненных учащимся на этапе подготовки к расчету. Это может быть оценка правильности выдвинутой гипотезы в учебном исследовании, оценка эффективности проектного решения, оценка качества построения математической модели и т.п. Кроме оценки, учащемуся может предоставляться и определенная вспомогательная информация для анализа и коррекции принятых решений. Степень развернутости этой информации, помогающей учащемуся принимать рациональные решения, определяется результатами оценки его деятельности.

Выделим ряд общих требований к внутренней ОС:

оперативность;

наглядность;

вариантность по степени оказания помощи;

продуктивно-творческий характер вспомогательной информации;

дружественная форма человеко-машинного диалога.

Существенный вклад в реализацию этих требований вносит применение в тренажерах интерактивной машинной графики. Известно, что скорость восприятия информации, представленной в графическом виде, на несколько порядков выше, чем скорость чтения и осмысления символьных данных. Говорят, что, например, инженеры мыслят образами. Поэтому применение машинной графики, особенно интерактивной, заметно интенсифицирует и повышает качество познавательной деятельности.

Требование продуктивно-творческого характера вспомогательной информации означает, что внутренняя обратная связь должна быть не в виде заранее подготовленных подсказок, разъяснений, как это часто бывает в электронных учебниках, а в виде такой информации об изучаемых объектах или процессах (преимущественно в наглядной графической форме), которая побуждала бы обучаемого к размышлению и рефлексии. Эта возможность появляется за счет математического моделирования, обеспечивающего высокую степень полноты и достоверности получаемых данных об изучаемых объектах или процессах.

При проектировании сценариев тренажеров для изучения объектов или процессов целесообразно использовать так называемые коэффициенты чувствительности, которые представляют собой частные производные какой-либо важной характеристики объекта или процесса по различным параметрам. Коэффициенты чувствительности являются сами по себе внутренней ОС высокого дидактического качества. Анализ коэффициентов чувствительности в ходе работы на тренажере по-

зволяет выявить активные и пассивные параметры, исследовать их влияние на характеристики изучаемого объекта или процесса.

Требование дружественной формы человеко-машинного диалога предполагает естественность языка диалога, наличие ободряющих реплик в лексиконе тренажера, быстрый отклик на запрос учащегося (не более 2-3 секунд задержки), наличие подсказок по технике ведения диалога. Словом, задачи могут быть сложными и трудными, но общение с тренажером должно быть простым и приятным для учащегося. Следует избегать фамильярности в репликах. "Живость" в положительных или отрицательных оценках, конечно, разнообразит диалог, но может неадекватно восприниматься разными людьми. Лучший способ сохранить хорошие отношения между тренажером и учащимся - это корректность и определенная "дистанция" в репликах тренажера.

Информация внешней ОС необходима преподавателю для анализа самостоятельной работы учащихся и коррекции всего процесса обучения. Вовсе не обязательно, чтобы она была оперативной. Анализ информации внешней ОС может быть отсроченным, а коррекция по его результатам может проводиться в ходе групповых и индивидуальных консультаций, в процессе формирования банка учебных задач, последовательности их предъявления учащимся, при совершенствовании тренажеров. Внешняя ОС должна предоставлять преподавателю возможность получать объективную количественную оценку учебной деятельности каждого ученика и статистику по учебной работе всей группы. Преподаватель должен иметь возможность анализировать не только итоговую оценку, но и путь, по которому продвигался учащийся в ходе решения учебной задачи. Такой анализ позволит оказывать более дифференцированную помощь при проведении индивидуальных консультаций.

1.3. Использование эвристических решений

Третий принцип построения сценариев тренажеров – обязательное эвристическое решение задач, предлагаемых при работе с тренажером, с последующим сопоставлением результатов с машинным вариантом решения. Проиллюстрируем этот принцип схемой типового сценария тренажера для проектировщиков. Эвристическое проектирование предполагает диалог с компьютером: учащийся генерирует варианты проекта, а компьютер проводит анализ предлагаемых вариантов и оценивает их по выбранному критерию эффективности. Заметим, что при "ручных" расчетах проанализировать много вариантов проекта невозможно. Быстрая же качественная оценка, не говоря уже о количественной оценке, недоступна порой даже опытному педагогу-

проектировщику. Применение компьютера позволяет автоматизировать трудоемкие рутинные вычисления и оставить за учащимся только те функции, которые требуют интеллекта, т.е. функции осмысления результатов и принятия решений. Количество анализируемых вариантов проекта резко увеличивается и вместе с тем увеличивается объем накопленных знаний об объекте или процессе проектирования при неизменном времени обучения.

Нередко рациональное проектное решение может быть получено и с помощью машинной оптимизации, например, с использованием методов нелинейного программирования. Однако и в этом случае эвристическое проектирование должно предшествовать машинной оптимизации. В ходе диалога учащегося с компьютером целесообразно предоставлять ему сначала лишь информацию о величине критерия эффективности оптимального проекта, чтобы активизировать процесс решения проектной задачи, а полную машинную оптимизацию давать возможность использовать лишь после выполнения определенного числа попыток эвристического проектирования. Такая последовательность учебной работы позволяет учащимся проявить свои творческие способности и в полной мере оценить достоинства, а порой и недостатки машинной оптимизации.

На завершающем этапе тренажа проектировщика целесообразно планировать анализ наиболее интересных и поучительных проектных задач рассматриваемого класса. Для этой цели в тренажере формируют специальный архив знаний. По каждой задаче в архиве хранят ее исходные данные и оптимальные решения в виде, допускающем различные формы представления результатов. Архив знаний может содержать также комментарии опытного преподавателя-проектировщика, которые могут предъявляться учащемуся по его запросу. Анализ оптимальных решений из архива знаний, подкрепляемый комментариями опытного преподавателя, позволяет закрепить и усилить учебный эффект предшествующего самостоятельного проектирования.

1.4. Состязательность в учебной работе

Четвертый принцип построения тренажеров – создание соревновательных ситуаций для активизации познавательной деятельности. Схема рассмотренного выше сценария учебного проектирования позволяет легко ввести соревновательные и, следовательно, игровые элементы. Это может быть соревнование либо на получение наиболее рационального проекта при выдаче одинаковых заданий всем учащимся, либо на достижение минимальной относительной разницы в критериях эффективности между эвристическими и оптимальными машин-

ными решениями при выдаче различных заданий. Причем при второй форме соревнования учащиеся непосредственно "состязаются" с компьютером, что, как показывают наблюдения, является психологически более щадящим и более привлекательным для большинства учащихся, чем прямое состязание друг с другом.

Конечно, не всегда соревновательная ситуация сама просится в сценарий тренажера, как в проектных задачах. Следовательно, ее необходимо придумать. Типовая соревновательная ситуация предполагает наличие какого-либо простого критерия оценки решения задачи и некоторого его конечного значения, к которому должен стремиться учащийся в процессе решения задачи. Даже такой простой игровой элемент удивительно стимулирует учебную работу.

Заключение

Общие принципы, рассмотренные выше, не охватывают всей совокупности методических способов и приемов построения сценариев интеллектуальных тренажеров [1-3]. Многие в них определяет специфика предметной области [4]. Однако указанные принципы разработки интеллектуальных тренажеров могут составить базис для разработки соответствующей методики по разработке электронных образовательных средств подобного типа, включающей следующие проектные этапы:

1. Выбор учебной задачи.
2. Разработка общей схемы (сюжета) сценария учебной работы.
3. Построение оценочной функции. Т.е. функции, по изменению которой обучающейся может судить о правильности своих действий в процессе решения учебной задачи.
4. Детализация сценария. На этом этапе полностью определяют способы деятельности учащегося. Разрабатывают диаграммы деятельности, которые дополняются их спецификациями – текстовым описанием взаимодействия объектов и последовательности действий с иллюстрациями в виде экранных форм. Такое описание позволяет одновременно с детализацией учебной деятельности определить форму человеко-машинного интерфейса тренажера.
5. Разработка функциональной схемы. Описание сценария позволяет определить основные функции тренажера и состав его подсистем.
6. Разработка программного обеспечения.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Соловов А.В. Электронное обучение: проблематика, дидактика, технология. – Самара: «Новая техника», 2006. – 454 с.
2. Новый подход к инженерному образованию: теория и практика открытого доступа к распределенным информационным и техническим ресурсам. Под редакцией А.А.Полякова. М.: Центр-Пресс, 2000. – 238 с.
3. Андрейчинков А.В., Андрейчинкова О.Н. Интеллектуальные информационные системы: Учебник. – М.: Финансы и статистика, 2006. – 424 с.
4. Техническое задание на ОКР «Разработка автоматизированной системы дистанционного обучения подготовки и повышения квалификации специалистов по спутниковой навигации». М.: ОАО «Российские космические системы», 2009. - 43 с.

Е.М. ДОНДИК, О.С. ВАВИЛОВА

Рязанский государственный радиотехнический университет

СИСТЕМА АНАЛИЗА ИСКАЖЕННЫХ ФОНЕМ

Рассматривается анализ искаженных фонем с помощью вейвлет преобразования на основе нейронных сетей.

Рассматриваемая система предназначена для обучения правильного произношению фонем слабослышащих и глухих детей, которые сами не могут оценивать качество своей речи. Поэтому возникает необходимость более точного распознавания и анализа искажений произносимых фонем. При этом качество произношения фактически должна оценивать техническая система, которая по своим характеристикам должна приблизиться к такому высококачественному анализатору, как человеческое ухо. Поэтому система должна быть максимально чувствительна к искажениям произносимых фонем для выдачи рекомендаций по корректировке произношения путем подстройки речевого аппарата. Для этого математический аппарат распознавания должен в определенной мере соответствовать реально происходящим процессам восприятия звуков человеческим ухом. Упрощенная структура системы приведена на рис. 1.

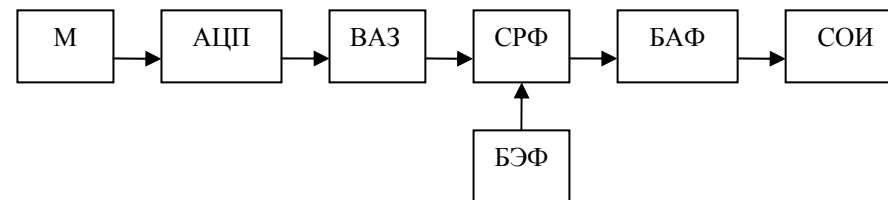


Рис. 1. Структурная схема системы распознавания и анализа искаженных фонем

Ввод звука в систему осуществляется с помощью микрофона (М), который преобразует звуковые колебания в электрический сигнал. С помощью аналого-цифрового преобразователя (АЦП) сигнал оцифровывается. Затем сигнал попадает на вейвлет анализатор звука (ВАЗ), в котором с помощью вейвлета Добеши 8 происходит обработка сигнала. На выходе ВАЗ формируется вектор из восьми коэффициентов, которые представляют собой средние значения энергий детализирующих коэффициентов, рассчитанные на каждом уровне вейвлет разложения. Система распознавания фонем (СРФ) представляет собой нейронную сеть встречного распространения, которая имеет 8 входов и 29 выходов. Обобщающая способность нейросети дает возможность получать правильный распознанный сигнал, при этом входной вектор неполон или искажен как в рассматриваемом случае. На вход нейросети поступает вектор из восьми коэффициентов, который сравнивается с векторами из банка эталонных фонем (БЭФ) и по правилу функционирования нейронных сетей встречного распространения на выходе остается активным лишь один нейрон, соответствующий в той или иной мере произнесенной фонеме. Затем в блоке анализа фонем (БАФ) происходит определение степени схожести произнесенного звука на образец, взятый из БЭФ. Процент идентичности произнесенного звука образцовому, отражается с помощью системы отображения качества произношения (СОИ), которая в зависимости от процента схожести звуков окрашивает символ изучаемой фонемы в определенный цвет, отображаемый процент схожести звуков

Психоакустические явления определяются физиологическим строением уха, состоящим из наружного, среднего и внутреннего уха.

Наружное ухо является звукоулавливающим аппаратом. Звуковые колебания улавливаются ушными раковинами и передаются по наружному слуховому проходу к барабанной перепонке, которая отделяет наружное ухо от среднего. Главное назначение ушной раковины человека состоит в защите слухового прохода. Барабанная перепонка представляет собой относительно жесткий конус, направленный внутрь, с углом у вершины около 135° , ее поверхность - около $0,8 \text{ см}^2$.

За барабанной перепонкой находится заполненная воздухом полость среднего уха. Звуковая волна проходит через наружное ухо и слуховой проход, вызывая колебание барабанной перепонки. Это колебание передается во внутреннее ухо. Акустико-механический импеданс внутреннего уха намного превышает импеданс воздуха, поэтому для эффективной передачи энергии звука требуется повышение импеданса.

Внутреннее ухо состоит из улитки, в нормальном состоянии свернутой в плоскую спираль с двумя с половиной оборотами и напоминающей раковину улитки, вестибулярного аппарата и окончаний слухового нерва. В улитке происходит преобразование механических процессов в нервные сигналы. Если улитку развернуть и вытянуть, она будет иметь вид, схематически показанный на рис. 2.

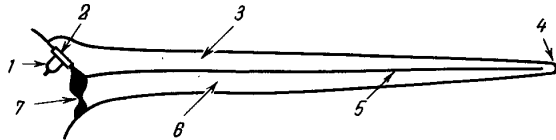


Рис. 2. Упрощенная схема развернутой улитки: 1 - стремечко, 2 - овальное окно, 3 — преддверная лестница, 4 - геликотрема, 5 - перегородка улитки, 6 - барабанная лестница, 7 - круглое окно.

Полость улитки заполнена бесцветной жидкостью перилимфой, вязкость которой примерно в два раза больше вязкости воды, а удельный вес примерно равен 1,03. Длина канала в спиральной раковине составляет примерно 35 мм. Площадь поперечного сечения у стремечка составляет приблизительно 4 мм^2 , она уменьшается примерно до 1 мм^2 у противоположного тонкого конца.

Очень медленные колебания стремечка, например, с частотой менее 20 гц, вызывают переливания жидкости из преддверной лестницы в барабанную и обратно через отверстие. Колебания более высокой частоты передаются через мягкую перегородку улитки, причем место передачи зависит от частоты звукового воздействия.

Главным элементом, определяющим основные функциональные динамические свойства перегородки, является мембрана, содержащая около 30 000 чувствительных клеток, к которым подходят окончания слухового нерва. Мембрана у основания уже и значительно жестче и тоньше, чем у вершины, где она более податлива и массивна. Поэтому резонансные свойства мембраны непрерывно изменяются вдоль ее длины. Различные участки мембраны являются своеобразными биомеханическими фильтрами, а мембрана в целом может рассматриваться

как набор фильтров, упорядоченных по частоте и покрывающих в совокупности всю область частот, доступных восприятию человека. Фильтры, более близкие к основанию улитки отвечают за более высокие частоты. Амплитудная и фазовая характеристики для некоторой заданной точки мембраны весьма схожи с соответствующими характеристиками полосового фильтра, обладающего относительно широкой полосой пропускания.

Механические движения мембраны превращаются в нервное возбуждение в органе, состоящем из большого числа клеток, среди которых имеются и волосковые клетки. Волоски, выходящие из этих чувствительных клеток, проникают сквозь сетчатую пластинку и соприкасаются с мембраной перегородки улитки. Число внутренних волосковых клеток составляет около 5000, число наружных волосковых клеток - около 25 000. Изгиб волосков вызывает электрические разряды в улиточной части нерва.

Чувствительные клетки уха соединены с мозгом пучком нервных клеток, или нейронов, образующим слуховой нерв. Общее число нейронов в слуховом нерве доходит, примерно, до 30 000. Нейроны имеют только два состояния: активное и заторможенное. При возбуждении входным электрическим сигналом, превышающим некоторый порог, нейроны генерируют стандартный электрический импульс длительностью около 1 мсек, после чего наступает период нечувствительности, длящийся от 1 до 3 мсек.

Человеческий слух во многом подобен спектральному анализатору, то есть, ухо распознаёт спектральный состав звуковых волн без анализа фазы волны.

Природа нашего слуха такова, что ухо раскладывает поступающие в него звуки на отдельные частоты. Звуковая волна, поступающая в ухо, преобразуется в колебания мембраны. Мембрана имеет разную жесткость по своей длине, и соответственно – разные собственные частоты колебаний разных участков. Когда на мембрану поступает сложное колебание, оно возбуждает колебания тех участков мембраны, которым соответствуют отдельные гармонические составляющие сложного входного колебания. Таким образом, результирующее преобразование сигнала с точностью до константы совпадает с вейвлет-преобразованием сигнала.

Вейвлеты - это обобщенное название особых функций, имеющих вид коротких волновых пакетов с нулевым интегральным значением и с той или иной, подчас очень сложной формой, локализованных по оси независимой переменной (t или x) и способных к сдвигу по ней и масштабированию (сжатию/растяжению).

Спектральное представление вейвлетов аналогично заданию окна в оконном преобразовании Фурье. Но отличие состоит в том, что свойства окна, его ширина и перемещение по частоте, присущи самим вейвлетам. Это служит предпосылкой их адаптации к сигналам, представляемым совокупностью вейвлетов. Поэтому нетрудно понять, что с помощью вейвлетов можно осуществить анализ и синтез локальной особенности речевого сигнала, который является частным случаем нестационарного сигнала.

Интегральным вейвлет-преобразованием функции $f(t) \in L^2(\mathbb{R})$ называется выражение

$$W(a, b) = \frac{1}{|a|^{1/2}} \int_{-\infty}^{\infty} f(t) \psi^* \left(\frac{t-b}{a} \right) dt, \quad (1)$$

где $a, b \in \mathbb{R}$, $a \neq 0$.

Функция $\psi(t)$ называется материнским вейвлетом. Символом $*$ обозначена процедура комплексного сопряжения. Параметр a определяет размер вейвлета и называется масштабом. Параметр b задает временную локализацию вейвлета и называется сдвигом.

Вейвлет-преобразование как и человеческое ухо обладает свойством фильтрации, такого преобразования исходной функции, которое приводит к изменению ее спектрального состава. Зададим фильтр функции $f(t)$ с помощью свертки

$$\phi(t) = \int_{-\infty}^{\infty} h(t') f(t-t') dt', \quad (2)$$

где $h(t)$ - весовая функция фильтра, удовлетворяющая условию

$$\int_{-\infty}^{\infty} h(t) dt = 1. \quad (3)$$

Преобразуя (2) по Фурье, получаем

$$\hat{\phi}(w) = \hat{h}(w) \hat{f}(w). \quad (4)$$

В этом равенстве $\hat{h}(w)$ - передаточная функция фильтра (2), которая является Фурье-образом весовой функции:

$$\hat{h}(w) = \int_{-\infty}^{\infty} h(t) e^{-iwt} dt. \quad (5)$$

Например, для ступенчатой весовой функции

$$h(t) = \begin{cases} \frac{1}{2T}, & |t| \leq T, \\ 0, & |t| > T \end{cases} \quad (6)$$

передаточная функция имеет вид

$$\hat{h}(w) = \frac{\sin(Tw)}{Tw}. \quad (7)$$

В общем случае из (3) следует $\hat{h}(0) = 1$. Таким образом, оператор (2) задает низкочастотную фильтрацию. Ширина пропускания передаточной функции (7) определяется только длиной весовой функции фильтра, то есть значением параметра T .

Сравнивая преобразования (1) и (2), мы видим, что интегральное вейвлет-преобразование (1) можно также считать фильтром исходной функции $f(t)$. Применяя к (1) преобразование Фурье по переменной b , получаем

$$\hat{W}(a, w) = \sqrt{a} \hat{\psi}^*(aw) \hat{f}(w), \quad (8)$$

откуда видно, что передаточная функция вейвлет-преобразования имеет вид

$$H(w) = \sqrt{a} \hat{\psi}^*(aw). \quad (9)$$

В силу

$$\hat{\psi}(0) = 0 \quad (10)$$

из (9) получаем

$$H(0) = 0 \quad (11)$$

Добавим также, что для каждого значения масштаба $a > 0$, максимум функции $H(w)$ лежит на частоте

$$w_{\max} = \frac{C}{a}, \quad (12)$$

где величина C зависит от взятого вейвлета. Эффективная ширина функции $H(w)$ убывает с ростом масштаба a . Другими словами, вейвлет-преобразование – это полосовой фильтр с переменной полосой пропускания, узкой для больших масштабов и широкой для малых.

Ранее было показано, что улитку внутреннего уха также можно представить как гребенку фильтров, выходной сигнал которой упорядочен по расстоянию от основания улитки со многими входами – по-

перечными волосками и выходами наружных волосковых клеток на нервные волокна. Следовательно, улитка передает в мозг вейвлет-образ речевого сигнала.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Фланаган Д.Л. Анализ, синтез и восприятие речи. Перевод с английского под редакцией Пироговой А.А. – М.: «СВЯЗЬ» 1968
2. Горшков Ю.Г. Новые решения речевых технологий безопасности // «Специальная техника» 2006 №4 стр.3-16
3. Найда С.А. Спектральный анализ звуков акустическим слуховым анализатором // Сборник акустического симпозиума «КОНСОНАНС-2005». – Киев, 2005. – С.145-150.

М.А. ЗОРИНА, Е.Д. НЕБЫКОВА

Рязанский государственный радиотехнический университет

РАЗРАБОТКА МОДУЛЯ ОБРАБОТКИ ОБРАЩЕНИЯ КЛИЕНТА ДЛЯ БАНКОВСКОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ С ИСПОЛЬЗОВАНИЕМ CRM-СТРАТЕГИЙ

Рассматривается разработка модуля взаимодействия с клиентами для банковской информационной системы с помощью CRM-стратегий. Преимущества использования CRM-стратегий.

В настоящее время российские предприятия наиболее активно используют привлечение долгового финансирования. Среди наиболее популярных инструментов стоит выделить кредиты банков. Выдача кредитов для банковской системы является сложнейшим процессом, сопряженным с получением, анализом и хранением информации о клиентах, с выбором продуктов кредитования и обработкой значительной информации на различных этапах сопровождения кредита.

Для ускорения и повышения качества данного процесса, возникла необходимость создания автоматизированной системы с использованием CRM-стратегий, позволяющей удовлетворить данным требованиям.

Система управления взаимодействием с клиентами (сокр. от англ. *Customer Relationship Management System*, CRM-система) - корпоративная информационная система, предназначенная для автоматизации CRM-стратегии компании, в частности, для повышения уровня продаж, улучшения обслуживания клиентов путём сохранения информации о клиентах и истории взаимоотношений с ними, установления и

улучшения бизнес-процедур и последующего анализа результатов. Этот подход подразумевает, что при любом взаимодействии с клиентом по любому каналу, сотруднику компании доступна полная информация обо всех взаимоотношениях с этим клиентом и решение принимается на основе этой информации (информация о решении, в свою очередь, тоже сохраняется).

CRM-стратегия — это формирование перспективных целей и наиболее эффективных методов их достижения, позволяющих получать конкурентные преимущества и дополнительную прибыль за счет внедрения в работу предприятия принципов главенствования потребностей клиента.

Для решения данной задачи использовалось новое программное обеспечение *Pivotal 5.9*.

В основе CRM-системы *Pivotal* – платформа, позволяющая создавать конфигурации, максимально соответствующие бизнес-задачам клиента. Платформа поставляется с любой конфигурацией продукта, что дает клиенту или партнеру возможность разработать собственное уникальное CRM-решение.

На рис.1 предложена архитектура CRM-системы для банковской информационной системы.

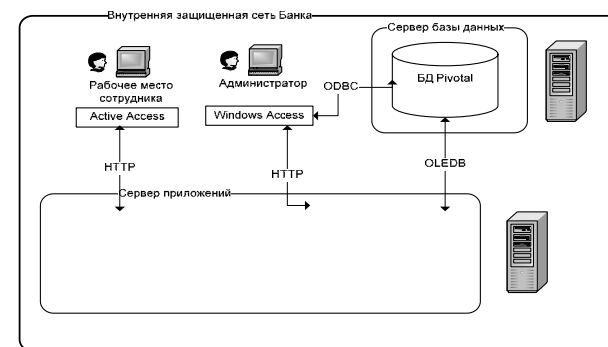


Рис. 1. Архитектура системы

Взаимодействие между сервером приложений и рабочими местами сотрудников должно происходить по протоколу *HTTPS (SSL)*. Для взаимодействия с сервером приложений Системы, сотрудники на своих рабочих местах должны использовать браузер *Internet Explorer* версии 6.0 и выше. Взаимодействие между *Pivotal* и сервером базы данных должно осуществляться по технологии *ODBC*. В качестве сервера базы данных выступает MS SQL Server 2005.

Обработка обращения клиента – один из этапов бизнес-процесса выдачи и сопровождения кредитов. Главная цель этого этапа – выявить заинтересованность Клиента в кредитном продукте, выбрать кредитную программу, проинформировать Клиента о требованиях Банка к заемщику и предоставить список необходимых документов, распределить обращения Клиентов между кредитными экспертами.

Модуль обладает главными качествами, которые нужны при работе с банковской информацией: безопасностью, защитой информации от несанкционированного доступа (используется аутентификация и авторизация пользователей), надежностью, шифрованием полей таблиц, содержащих информацию субъекта персональных данных, сохранением целостности информации в ее базах данных. При помощи встроенных средств *Pivotal* достигаются эти требования.

Результатом разработки CRM-системы выдачи и сопровождения кредитов являются следующие основные преимущества:

- повышение оперативности принятия решений;
- снижение трудозатрат сотрудников банка на получение информации по кредитам;
- снижение вероятности ошибок сотрудников банка при операциях с кредитами.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Павел Черкашин, Лаборатория знаний, Интернет-университет информационных технологий, Бином. 376 с.
2. Пол Гринберг, CRM со скоростью света: привлечение и удержание клиентов в реальном времени. - СПб. Символ-Плюс, 2006. 528с.
3. <http://www.pivotal.com> Дата просмотра 15 апреля 2010г.

П.В. КАРМАНОВ

Рязанский государственный радиотехнический университет

ДИСКРЕТНЫЙ ФАЗОВРАЩАТЕЛЬ

Рассматривается дискретный фазовращатель (ФВ), обеспечивающий получение высоких точностных характеристик радиотехнических и измерительных устройств.

Дискретный фазовращатель с высокими точностными характеристиками можно построить на счетных Т-триггерах, собранных на элементах И-НЕ по способу трех асинхронных триггеров R-S типа

[1,2]. Такое построение позволяет получить от Т-триггера не только сдвинутые по фазе на 180^0 друг относительно друга выходные сигналы: «Выход 0^0 » и «Выход 180^0 », но и выходной сигнал, задержанный относительно сигнала «Выход 0^0 » на длительность, равную длительности входного импульса τ . При запуске триггера сигналом формы «меандр» этот дополнительный выход имеет задержку в 90^0 . С помощью дополнительной схемы НЕ можно получить еще один выходной сигнал с задержкой относительно сигнала «Выход 0^0 » на 270^0 .

Таким образом, каждый счетный триггер с дополнительной схемой НЕ позволяет получить выходные сигналы формы «меандр», сдвинутые друг относительно друга на 90^0 : «Выход 0^0 », «Выход 90^0 », «Выход 180^0 » и «Выход 270^0 ».

Структурная схема фазовращателя для $N=3$ (N - количество Т-триггеров в цепи деления частоты) представлена на рис.1.

Запускающие импульсы типа «меандр» поступают на первый Т-триггер, он является запускающим для ФВ. «Выход 90^0 » этого триггера запускает цепь последовательного деления частоты (фазы) сигнала, состоящую из $(N-1)$ последовательно соединенных триггеров (триггеры пять и шесть). Каждый последующий триггер запускается сигналом «Выход 0^0 » предыдущего триггера.

На выходе цепи деления частоты формируется сигнал, имеющий минимальный фазовый сдвиг $\Delta\phi_d$, равный

$$\Delta\phi_d = 2\pi / (4 \cdot 2^{N-1}) = \pi / 2^N, \quad (1)$$

где $\Delta\phi_d$ – требуемый дискрет фазового сдвига ФВ.

Исходя из заданного значения $\Delta\phi_d$, можно определить количество триггеров $(N-1)$ этой цепи.

Сигнал «Выход 0^0 » первого триггера запускает другую цепь деления частоты (триггеры 2 и 3).

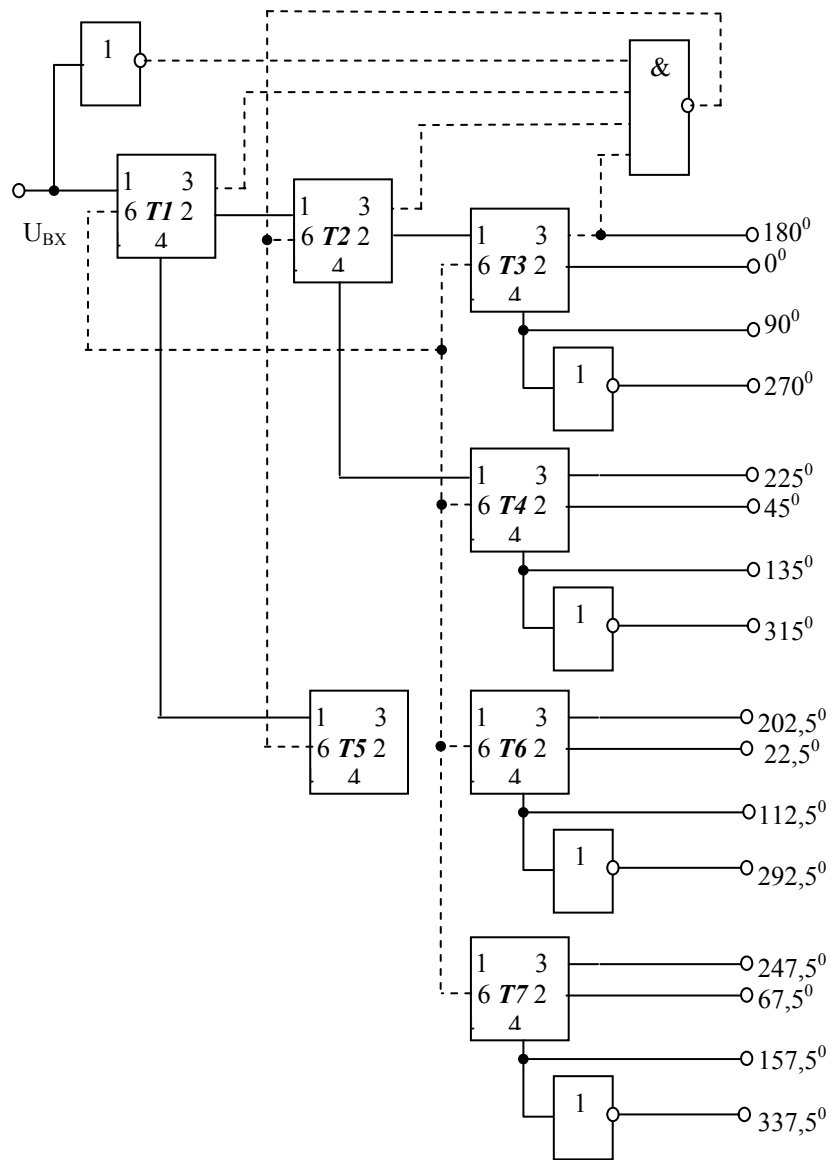


Рис.1. Структурная схема дискретного фазовращателя

Каждый триггер описанных двух цепей деления частоты запускает сигналом «Выход 0^0 » последующий триггер, а сигналом «Выход 90^0 » - дополнительную цепь делителя частоты.

В каждой дополнительной цепи сигнал «Выход 0^0 » предыдущего триггера запускает последующий триггер, а сигнал «Выход 90^0 » - дополнительную цепь деления частоты. Исходя из такого построения ФВ, получаем, что любая цепь последовательного деления частоты сигнала, считая с первого триггера и кончая выходным триггером, состоит из N триггеров.

Все выходные узлы (последний триггер и элемент НЕ) имеют одновременно выходные сигналы одной частоты

$$f_{\text{вых}} = f_{\text{вх}} / 2^N, \quad (2)$$

где $f_{\text{вх}}$ - частота входного сигнала первого триггера ФВ.

Фазовые сдвиги сигналов отличаются от сигнала «Выход 0^0 » по фазе на $\Delta\phi_d$, $2\Delta\phi_d$ и т.д. в пределах от $0^0 \div 360^0$. Общее количество выходов ФВ равно

$$p = 2\pi / \Delta\phi_d = 2^{N+1}. \quad (3)$$

Количество триггеров, имеющих одну и ту же выходную частоту (на рис.1. они расположены в одной вертикали) определяется из выражения

$$Q = 2^N - 1. \quad (4)$$

В схеме ФВ (рис.1.) цепи синхронизации представлены пунктирными линиями.

Погрешность при формировании фазового сдвига выходного сигнала получается наибольшая в той цепи деления частоты, которая состоит из последовательно соединенных триггеров, вход каждого из которых подключен к цепи «Выход 90^0 » предыдущего триггера (триггеры 1,5,7 рис.1.).

В каждом триггере «Выход 90^0 » относительно сигнала «Выход 0^0 » имеет задержку $\Delta\tau$, равную задержке срабатывания одной микросхемы И-НЕ, поэтому задержка выходного сигнала для этой цепи деления равна

$$\delta\phi_1 = \Delta\tau \cdot N, \quad (5)$$

где $\delta\phi_1$ - максимальная задержка сигнала в цепи деления в секундах; $\Delta\tau$ - задержка срабатывания одной микросхемы.

Фазовый угол, формируемый в этой цепи, определяется из выражения

$$\phi_1^0 = 90^0 (1/2^0 + 1/2^1 + \dots + 1/2^{N-1}) = 180^0 (2^N - 1) / 2^N \quad \text{или} \quad \phi_1(c) = T_{\text{вых}} (2^N - 1) / 2^{N+1}. \quad (6)$$

Относительная погрешность формирования фазового сдвига для данной цепи равна

$$A_1 = \delta\varphi_1 / \varphi_1 = \Delta\tau \cdot N \cdot 2^{N+1} / T_{\text{ВЫХ}} (2^N - 1). \quad (7)$$

Задержка выходного сигнала в цепи, задающей минимальный фазовый сдвиг (триггеры 1,5,6 рис. 1.) равна $\Delta\varphi_2 = \Delta\tau$. В этой цепи фазовый сдвиг равен заданному дискрету сдвига ФВ. $\varphi_2^0 = \Delta\varphi_{\text{д}} = \pi / 2^N$

или $\varphi_2(\varepsilon) = T_{\text{ВЫХ}} / 2^{N+1}$.

Относительная погрешность формирования фазового сдвига для данной цепи равна

$$A_2 = \delta\varphi_2 / \varphi_2 = \Delta\tau \cdot 2^{N+1} / T_{\text{ВЫХ}}. \quad (8)$$

Период входного сигнала ФВ равен $T_{\text{ВХ}} = T_{\text{ВЫХ}} \cdot 2^N$.

Таким образом, дискретный ФВ имеет следующие преимущества:

- возможность полного построения ФВ на цифровых микросхемах, что определяет его высокую надежность и малые габариты;
- высокая точность формирования фазового сдвига;
- возможность одновременного получения выходных сигналов, сдвинутых по фазе на $\Delta\varphi_{\text{д}}$ друг относительно друга в пределах $0^\circ \div 360^\circ$.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Букреев И.Н. и др. Микроэлектронные схемы цифровых устройств. М.: Радио и связь, 1990. 416 с.
2. Шляпоберский В.И. Основы техники передачи дискретных сообщений. М.:Связь, 1973, 480 с.

П.В. КАРМАНОВ

Рязанский государственный радиотехнический университет

ЦИФРОВОЕ МОДЕЛИРОВАНИЕ ИМПУЛЬСНОГО КВАДРАТУРНО-ФАЗОВОГО ДИСКРИМИНАТОРА ПРИ СЛУЧАЙНОМ ВХОДНОМ СИГНАЛЕ

Получены соотношения для выбора интервала дискретизации и длины реализации случайного процесса при цифровом моделировании импульсного квадратурно-фазового дискриминатора. Рассмотрены особенности алгоритма, описывающего процессы преобразования случайного сигнала в звеньях дискриминатора.

В современных системах приема частотно-модулированных сигналов широко применяются импульсные дискриминаторы (ИД) квад-

ратурно-фазового типа [1], импульсная форма выходного сигнала рас-согласования которых позволяет легко осуществить его цифровое интегрирование и запоминание кода частоты следящего гетеродина при одновременном уменьшении инструментальных ошибок измерения частоты случайного сигнала.

Структурная схема ИД приведена на рис. 1.

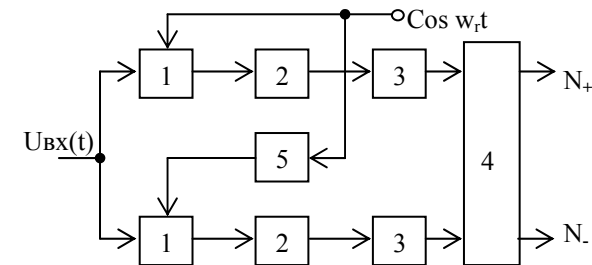


Рис.1. Структурная схема импульсного дискриминатора:

1- смесители; 2- фильтры низкой частоты; 3- усилители-ограничители; 4- формирователь импульсов расстройки; 5- фазовращатель на $\pi/2$.

Величина фазового дискрета $\Delta\varphi_{\text{д}}$ ИД зависит от количества каскадов совпадения формирователя импульсов расстройки равной $\pi/2$; π ; $0,5\pi$ соответственно при одном, двух и четырех импульсах рассогласования на один период частоты расстройки.

Цифровая модель импульсного дискриминатора

Цифровая модель дискриминатора, применяемая для экспериментального исследования на ЭВМ характеристик ИД, содержит следующие блоки, реализующие требуемые функциональные преобразования случайного сигнала:

- блок формирования формы спектра входного случайного сигнала и квадратурного перемножения его на сигнал гетеродина, расстроенный на некоторую частоту $\Delta F_{\text{р}}$ относительно средней частоты входного сигнала; при этом должно быть предусмотрено введение погрешности $\Delta\varphi_{\text{вч}}$ фазового сдвига квадратурного преобразования сигнала гетеродина;

- блок суммирования квадратурных компонент сигнала с аддитивным нормальным шумом; при этом должно быть обеспечено установление требуемого отношения максимальной спектральной плотности сигнала и шума $h=G_{\text{с}}/G_{\text{ш}}$;

- блок фильтрации низкочастотных компонент суммы сигнала и шума; при этом должна быть предусмотрена установка различных постоянных времени T фильтров низкой частоты (ФНЧ) с целью имитации различия фазовых сдвигов $\Delta\varphi_{\text{НЧ}}$ в квадратурных каналах ИД;
- блок логических операций формирования выходной последовательности импульсов рассогласования $N_+[n]$ и $N_-[n]$; при этом должна быть предусмотрена возможность изменения величины фазового дискрета $\Delta\varphi_{\text{д}}$;
- блок регистрации и обработки исследуемых характеристик ИД и параметров контрольного распределения интервалов.

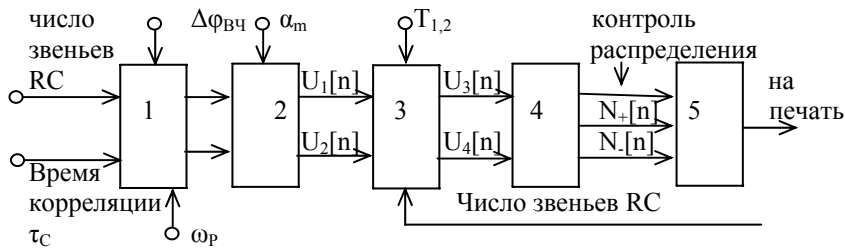


Рис. 2. Структурная схема цифровой модели импульсного дискриминатора:

1 – блок формирования сигнала и квадратурное перемножение; 2 – блок суммирования сигнала и шума; 3 – блок низкочастотной фильтрации; 4 – блок формирования импульсов расстройки; блок обработки характеристик процесса.

Случайный сигнал в цифровой модели ИД формируется с помощью двух низкочастотных независимых и нормально распределенных квадратурных компонент $A_c[n]$ и $A_s[n]$ его огибающей. Форма спектральных характеристик $A_c[n]$ и $A_s[n]$ при моделировании задается путем изменения числа рекурсий m алгоритма, реализующего форму спектра однозвенного RC-фильтра и не имеющего методической погрешности [2].

$$\xi[n] = x[n]\sqrt{1-\rho^2} + \rho\xi[n-1], \quad (1)$$

где $\xi[n]$ - значение случайного сигнала на n -м шаге ЭВМ;

$x[n]$ - случайное число с нормальным распределением в диапазоне от $-0,5$ до $+0,5$, получаемое по стандартной программе ЭВМ;

$\rho = \exp(-\tau_{\text{д}} / RC)$ - коэффициент корреляции случайного процесса на входе однозвенного RC-фильтра, взятый через интервал дискретизации $\tau_{\text{д}}$;

RC- постоянная времени RC-фильтра.

Аддитивный широкополосный шум задается путем прибавления к сигналу - непосредственно перед операцией низкочастотной фильтрации – независимых случайных и нормально распределенных чисел в каждом канале модели.

С учетом отмеченных особенностей цифровой модели ИД для входных сигналов блока фильтрации низкочастотных компонент суммы сигнала и шума будут справедливы соотношения

$$U_1[n] = A_c[n] \cos \omega_p[n] + A_s[n] \sin \omega_p[n] + U_{m1}[n];$$

$$U_2[n] = -A_c[n] \sin \{\omega_p[n] + \Delta\varphi_{\text{ВЧ}}\} + A_s[n] \cos \{\omega_p[n] + \Delta\varphi_{\text{ВЧ}}\} + U_{m2}[n]; \quad (2)$$

где $U_{m1}[n]$ и $U_{m2}[n]$ – независимые нормально - распределенные случайные числа;

ω_p - расстройка частоты сигнала и гетеродина.

Блок фильтрации $U_{m1}[n]$ и $U_{m2}[n]$ при моделировании описывается также с помощью алгоритма (1), что позволяет упростить моделирование в целом. При этом требуемое число RC- звеньев ФНЧ можно вводить в программу цифровой модели путем изменения числа рекурсий данного алгоритма.

Обозначим через $U_3[n]$ и $U_4[n]$ сигналы на выходе блока фильтрации сигналов $U_1[n]$ и $U_2[n]$, тогда алгоритм формирования импульсов рассогласования $U_+[n]$ и $U_-[n]$ для величины фазового дискрета $\Delta\varphi_{\text{д}}$ ИД, равной 2π , будет иметь вид:

$$N_+[n] \vee N_-[n] = 0, \text{ если } \text{sign}U_3[n] = \text{sign}U_3[n-1];$$

$$N_+[n] = 1, \text{ если } U_3[n]>0; U_3[n-1]<0; U_4[n]>0;$$

$$N_-[n] = 1, \text{ если } U_3[n]<0; U_3[n-1]>0; U_4[n]>0. \quad (3)$$

Аналогичным образом происходит формирование импульсов рассогласования и при $\Delta\varphi_{\text{д}}=\pi$ или $0,5\pi$.

Выбор интервала дискретизации и длины процесса при цифровом моделировании

При исследовании характеристик ИД критерием выбора интервалов дискретизации является точность отсчета моментов пересечения сигналом нулевого уровня.

Погрешность оценки интервалов между нулями может быть двух видов.

Первый вид погрешности связан с потерями некоторого числа интервалов в момент времени, когда в исходном непрерывном процес-

се происходит два или более пересечений нулевого уровня внутри одного дискрета $\Delta\tau_d$. второй вид погрешности связан с невозможностью отсчета длительностей интервалов с точностью, большей τ_d .

В качестве контрольного критерия для оценки правильности функционирования цифровой модели ИД выберем распределение интервалов между нулями низкочастотных сигналов на выходе смесителей ИД.

Для случайного процесса на выходе формирующего двухзвенного RC – фильтра распределение интервалов хорошо аппроксимируется экспоненциальным законом

$$W(t) = \frac{1}{\pi\tau_1} \exp\left(-\frac{t}{\pi\tau_1}\right) \quad (4)$$

с математическим ожиданием $m_x(t) = \pi\tau_1$ (где $\pi\tau_1 = RC$ – постоянная времени одного звена RC-фильтра).

Определим в пуассоновском приближении вероятность p потери некоторого числа k , равного двум или более интервалам пересечения случайным сигналом нулевого уровня:

$$p(k \geq 2) = 1 - \exp\left(-\frac{\tau_d}{\pi\tau_1}\right) - \frac{\tau_d}{\pi\tau_1} \exp\left(-\frac{\tau_d}{\pi\tau_1}\right) \quad (5)$$

С учетом малости величины $\tau_d / \pi\tau_1$ вероятность $p(k \geq 2)$ может быть выражена в виде

$$p(k \geq 2) \cong \tau_d^2 / (\pi\tau_1)^2. \quad (6)$$

Для оценки погрешности второго вида по-прежнему будем полагать поток пересечений нуля пуассоновским процессом, следовательно интервалы между ними будут распределены по показательному закону с интегральной функцией распределения

$$F_1(t) = 1 - \exp(-t / \pi\tau_1). \quad (7)$$

Связь между непрерывным исходным распределением интервалов $W(t)$ и дискретным $W_d(t)$ установим на основе приведенных ниже соотношений:

$$W_d(t) = \sum_{n=1}^{\infty} p_n(\tau_d) \delta(t - n\tau_d), \quad (8)$$

где n – номер машинного шага;

$\delta(t - n\tau_d)$ – дельта – функция Дирака;

$$p_n(\tau_d) = \int_{(n-1)\tau_d}^{n\tau_d} W(t) dt = F_1\{n\tau_d\} - F_1\{(n-1)\tau_d\}. \quad (9)$$

Для математического ожидания $m_d(t)$ дискретного распределения $M_d(t)$ справедливо соотношение

$$m_d(t) = \sum_{n=1}^{\infty} P_n(\tau_d) n\tau_d. \quad (10)$$

С учетом соотношений (7) и (9) выражение (10) можно записать следующим образом:

$$m_d(t) = \tau_d [\exp Q - 1] \sum_{k=1}^{\infty} k \exp(-kQ), \quad (11)$$

где $Q = \tau_d / \pi\tau_1$.

Сумму членов ряда в выражении (11) можно определить, используя известное соотношение для членов ряда арифметико-геометрической прогрессии

$$\sum_{k=0}^{\infty} k \exp(-kQ) = \exp(-Q) [1 - \exp(-Q)]^{-2}. \quad (12)$$

Подставив соотношение (12) в выражение (11), в котором функция $\exp(-Q)$ разложена в степенной ряд с точностью до членов второго порядка малости, имеем

$$m_d(t) = \pi\tau_1 (1 + \tau_d / 2\pi\tau_1). \quad (13)$$

Из выражения (13) следует, что при выборе отношения $\tau_1 / \tau_d \geq 5$, погрешность второго вида, отнесенная к величине $m_x(t) = \pi\tau_1$ непрерывного распределения, не превышает $3,2 \cdot 10^{-2}$, а вероятность потерь интервалов $p(k \geq 2)$ при этом составит величину, приблизительно равную $1 \cdot 10^{-3}$.

Флюктуационная ошибка измерения среднего значения $m_d(t)$ определяется соотношением

$$\alpha_m = m_d(t) / \sqrt{L-1}, \quad (14)$$

где L – число интервалов в реализации процесса длины T_p .

Таким образом, для того, чтобы флюктуационная ошибка α_m была не более величины ошибки второго вида, длина реализации T_p должна быть не менее $1,57 \cdot 10^4 \tau_1$.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. А.с. СССР № 748797. Импульсный частотный дискриминатор/ П.В. Карманов и др. 1980. Бюл. №26.
2. Быков в.В. цифровое моделирование в статистической радиотехнике. – М.: Сов. Радио. 1971. 328 с.

Д.И. КИЙ, К.Е. НЕКРОТОВ, В.Н. РУЧКИН, В.А. ФУЛИН
Рязанский государственный университет имени С.А.Есенина

УПРАВЛЕНИЕ ОБРАЗОВАТЕЛЬНЫМИ РЕСУРСАМИ ПОСРЕДСТВОМ СОВРЕМЕННЫХ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

Рассматриваются способы формирования электронных образовательных ресурсов с использованием современного программного продукта «Web Constructor».

В современных условиях экономического кризиса инновационными средствами являются ускоренное формирование экономики, основанной на знаниях, является управлением знаниями («Knowledge Management», или КМ) и рынке знаний. Причем историческое развитие показывает, что резкий скачок наблюдается в тех странах, где используют технологии управления знаниями.

Сказанное поднимает страну на новый уровень экономического развития и повышает ее конкурентоспособность посредством перехода к обществу знаний и стимулирования различных средств формирования, управления и рынка знаний.

Поэтому достаточно актуальна проблема создания электронных образовательных ресурсов, например, УМК* знаний или различных тренажеров по различным дисциплинам с возможностью доступа по технологиям e-Learning или m-Learning.

Кроме того, использование звука, графики, анимации и гипертекста существенно повышает качество учебных материалов, стимулирует участие молодежи в трудовой деятельности, существенно увеличивает производительность процесса обучения, удешевляет обучение.

В динамично меняющемся современном мире, на смену одним технологиям приходят другие. Перевод книг и журналов в цифровой вид уже давно повсеместно осуществляется. Пришло время классическим системам преподавания потесниться, и уступить место новым интерактивным и мультимедийным системам ведения и управления учебными процессами.

Для простого и понятного создания электронных образовательных ресурсов авторы статьи разрабатывают программный продукт под названием «Web Constructor». Он представляет собой двухуровневую систему создания и редактирования электронных учебников.

В режиме пользователя человеку достаточно дать ответ на небольшое количество простых вопросов, а программа посредством встроенной экспертной системы сама строит структуру ресурса. Поль-

зователю остается лишь наполнить учебник контентом — и полноценный, грамотно построенный образовательный сайт готов. Кроме того, имеется возможность добавить на сайт систему тестирования с встроенным дистанционным контролем — студент проходит тестирование, получает результат, а преподавателю приходит полная сводка об успеваемости студентов.

Режима пользователя содержит огромное количество встроенных шаблонов, цветовых решений, стилей, вариантов интерактивности, контролируемых режимов, а также очень гибкую систему выбора нужного варианта. Эта система основана на подробной и продуманной экспертной системе и позволяет пользователю получить именно тот вид своего ресурса, который он хочет. Кроме того, имеется возможность добавить собственные шаблоны, цветовые схемы или стили, а также настраивать содержимое этих компонентов вручную.

В режиме эксперта можно отредактировать уже созданный сайт при помощи инструментов визуального редактирования либо вручную поменять исходный код веб-документа, а также добавлять к ресурсу собственные модули, макросы и процедуры.

Основные компоненты режима эксперта:

2. Main menu (главное меню) — главное меню программы, в котором производится общее управление проектом, внешним видом рабочего окна, наборами инструментов и т.д. Workspace (рабочая область) - в нее входят:
3. Page view (просмотр страницы) — позволяет производить предпросмотр страницы;
4. Structure (структура) - отображает структуру текущей веб-страницы; Properties (свойства) — отображает свойства текущего объекта; Project View (окно проекта) - отображает дерево файлов и каталогов, входящих в данный проект;
5. Tags (тэги) — содержит полный набор компонентов, которые можно поместить на веб-страницу.

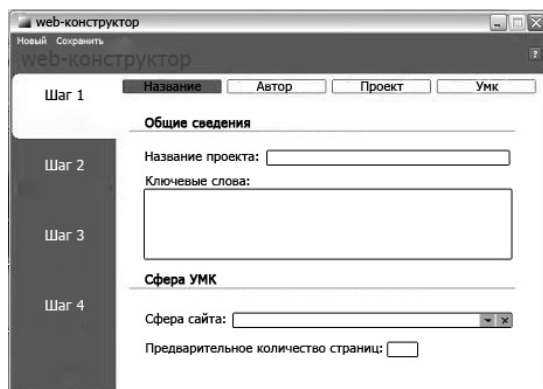


Рис. 1. Общий вид пользовательского режима

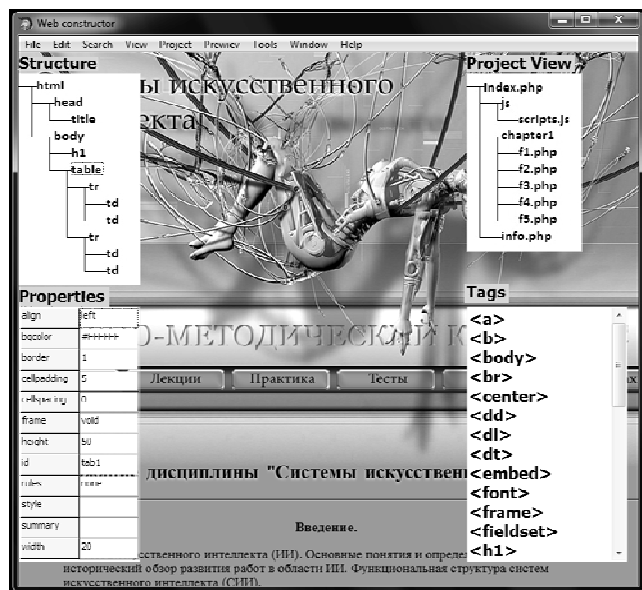


Рис. 2. Общий вид режима эксперта

Благодаря перечисленному набору инструментов ССУ УМК «Web Constructor» любой пользователь способен обучить, организовать и построить электронный образовательный ресурс в виде, например, учебно-методического комплекса, не имея специальных навыков в среде веб-программирования.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Технологии e-Learning и m-Learning. Журнал e-Learning World, №5 Октябрь – ноябрь 2006.
2. Электронные образовательные ресурсы. УМК «Системы искусственного интеллекта» <http://www.rsu.edu.ru/~d.kiy>
3. Системы создания и управления учебно-методическими комплексами.

М.К. КОЗЛОВ, Н.В. КРАВЧУК

Рязанский государственный радиотехнический университет

РАЗРАБОТКА ВЕБ-СЕРВИСА ДЛЯ ЛОКАЛЬНОГО ПРОСМОТРА ВЕБ-САЙТОВ

В данной работе разрабатывается веб-сервис - программная система, идентифицируемая строкой URI, чьи общедоступные интерфейсы определены на языке XML.

В нынешние времена бурного развития интрнета все чаще и чаще возникает проблема связи между данными, расположенными в разных концах земного шара на различных платформах и в разнообразных источниках данных.

Веб-сервис - программная система, идентифицируемая строкой URI, чьи общедоступные интерфейсы определены на языке XML. Описание этой программной системы может быть найдено другими программными системами, которые могут взаимодействовать с ней согласно этому описанию посредством сообщений, основанных на XML, и передаваемых с помощью интернет-протоколов.

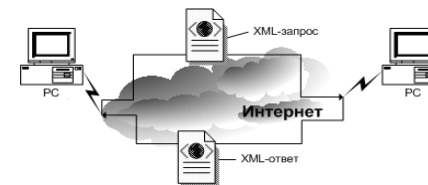


Рис. 1. Обмен данными в формате XML между приложениями по сети

Для чего же нужен данный сервис?

- Возможность в любое время сохранить полезную информацию с web-ресурса.

- Возможность поиска информации на сайте.
- Возможность фильтрации содержимого (html, jpg, png, mp3, flash), тем самым снижение расходов на трафик во время загрузки.

Доступ к сервису может осуществляться независимо от типа устройства, единственное условие — наличие браузера.

Сохранение информации и ее сжатие:

- Информация о пользователях храниться в MySQL
- Файловое хранилище MongoDB (GridFS)
- Сжатие осуществляется по выбору – tar.gz или zip

Почему MongoDB:

- Документо-ориентированное хранилище (простая и мощная JSON-подобная схема данных)
- Собственная файловая система – GridFS, эффективное хранение двоичных данных различных объёмов.
- Полная поддержка индексов
- Динамические запросы

Архитектура веб-службы базируется на следующих технологиях: WSDL — язык описания веб-сервисов, основанный на языке XML. SOAP — протокол обмена структурированными сообщениями в распределённой вычислительной среде. UDDI — инструмент для расположения описаний веб-сервисов (WSDL) для последующего их поиска другими организациями и интеграции в свои системы.

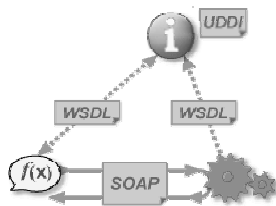


Рис. 2. Архитектура веб-службы

SOAP позволяет создавать приложения, удаленно вызывая методы объектов. SOAP устраняет требование того, что две системы должны запускаться на одной платформе или быть написаны на одном языке

программирования. Вместо вызова методов по определенному бинарному протоколу, пакет SOAP использует XML, основанный на тексте синтаксис для осуществления вызова методов. Вся информация между запрашивающим приложением и принимающим объектом пересылается в виде данных, заключенных между тегами, в XML-потоке по HTTP. С точки зрения Web-сервисов SOAP может быть реализован в качестве как клиента, так и сервера.

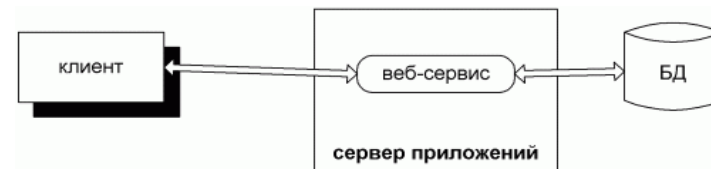


Рис. 3. Реализация сервиса

Для использования веб-сервиса клиент должен обладать следующей информацией:

- адрес, по которому расположен сервис
- имя сервиса
- метод и его параметры

Преимущества веб-сервиса:

- Для создания задания нужен только браузер.
- Экономия сетевого трафика.
- Веб-службы обеспечивают взаимодействие программных систем независимо от платформы

Веб-службы основаны на базе открытых стандартов и протоколов. Благодаря использованию XML достигается простота разработки и отладки веб-служб.

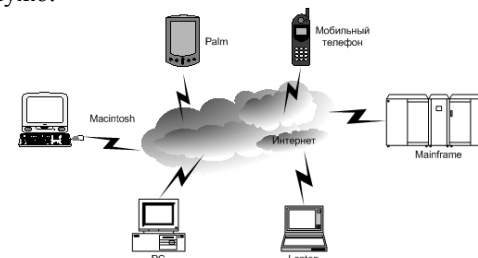


Рис. 4. Пример доступа к системы

Доступ к сервису может осуществляться независимо от типа устройства, единственное условие — наличие браузера. Проведен-

ный в рамках данной работы анализ показал, что задача оперативного локального доступа к контенту веб-ресурсов является актуальной, а средства, решающие ее востребованными. Одним из эффективных инструментов решения данной задачи является разработанный веб-сервис.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. <http://www.ibm.com>.
2. <http://www.apache.org>.
3. <http://www.mongodb.org>.
4. Г. Шилдт, Д. Холмс, Искусство программирования на JAVA.

А.В. КРОШИЛИН

Рязанский государственный радиотехнический университет

ПРЕДВАРИТЕЛЬНАЯ ПОДГОТОВКА МАССИВОВ ДАННЫХ ДЛЯ КЛАСТЕРИЗАЦИИ С ИСПОЛЬЗОВАНИЕМ ТЕХНОЛОГИИ TEXT MINING

Рассматривается применение технологии Text Mining при предварительной подготовке массива данных для проведения кластеризации. Для построения модели данных предлагается использовать семантические сети.

Во многих организациях, в настоящее время, в результате автоматизации своей деятельности при использовании баз данных скопилось большие объемы данных, в которых заключено большое количество дополнительной невыявленной и потенциально полезной информации [1]. Эффективный мониторинг накопленной статистической информации позволяет, например, определять статистические показатели для выявления и оценки существующих и потенциальных угроз неблагоприятных ситуаций, и подготовить мотивационную базу для принятия управленческих решений [2]. Эффективный мониторинг данных достигается путем применения методов интеллектуального анализа, особое место, в которых занимают методы кластеризации.

Кластеризация, основываясь на установленном отношении схожести элементов, устанавливает подмножества (кластеры), в которые группируются входные данные. Для успешной кластеризации массивов обрабатываемых данных необходима предварительная подготовка, которая заключается в выборе множества данных для анализа и выделении атрибутов, по которым будет производиться анализ. Эти атрибуты должны кратко и полно описывать исследуемое множество дан-

ных, остальные же, как не несущие полезной для анализа информации, исключаются из рассмотрения. Для решения этой задачи воспользуемся технологией Text Mining [3]. Text Mining - это набор технологий и методов, предназначенных для извлечения информации из текстов. Основная цель - возможность работать с большими объемами исходных данных за счет автоматизации процесса извлечения нужной информации.

В этой технологии процесс выбора множества данных для анализа можно условно разделить на три фазы. Сначала находятся релевантные массивы данных, потом из результатов поиска, извлекаются данные для предварительного анализа и в завершение их анализируем. При автоматизация извлечения информации из массива выделяется нужные элементы, при этом на вход подается набор информации, написанный на естественном языке, а на выходе получается информация в структурированном виде. Структуры могут представлять собой как простые сущности, так и сложные (факты, содержащие некое событие, его участников). События бывают самые разные. Указанный инструмент позволяет автоматически собирать результаты в коллекции данных, которые уже пригодны для проведения анализа.

Эту задачу Text Mining возможно решить, применяя модели данных, построенных на семантических сетях в которых извлечения ассоциативных правил из текстовых данных сформулированы на концептуальных графах с применением теории нечетких множеств [4]. При этом мера близости концептуальных графов полнее отображает семантическую близость текстовых данных. Создание семантической сети большого текста, подготовка резюме текста, поиск по тексту и автоматическая классификация и кластеризация текстов. Построение семантической сети - это поиск ключевых понятий текста и установление взаимоотношений между ними. По такой сети можно не только понять, о чем говорится в тексте, но и осуществить контекстную навигацию. Такой подход позволит качественней подготовить массив данных для дальнейшей кластеризации и позволяет получить дополнительные сведения для анализа информации. Это дает возможность осуществлять эффективное управление, составлять отчеты, графики, диаграммы и документы на основе постоянно динамически изменяющейся информации.

Применение нечеткой кластеризации позволяет увеличить объем обрабатываемой статистической информации, что в значительной степени повышает эффективность мониторинга в медицинских учреждениях.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Крошилина С.В. Разработка и исследование автоматизированных систем аналитики деятельности предприятия. Диссертация на соискание ученой степени кандидата технических наук - Рязань: РГРТА, 2009. – 169с.
2. Дюк В.А. «Интеллектуальный анализ данных» - СПб.: Питер, 2008.
3. Michael W. Berry. Survey of Text Mining. Clustering, Classification, and Retrieval. - Springer-Verlag, 2004. - 244 p.
4. Крошилин А.В. Разработка и анализ интеллектуальных поисковых программ в вычислительных сетях на основе универсальных алгебр. Диссертация на соискание ученой степени кандидата технических наук - Рязань: РГРТА, 2003. – 167 с.

М. В. КУРДЮМОВА

Рязанский государственный радиотехнический университет

СПОСОБЫ УВЕЛИЧЕНИЯ ПРОИЗВОДИТЕЛЬНОСТИ ВЕБ-ПРИЛОЖЕНИЙ

В статье рассматриваются подходы к разработке веб-приложений, при которых обеспечивается оптимальная скорость их загрузки.

С каждым годом веб-приложения становятся больше по размеру и сложнее по наполнению. Увеличиваются объемы загружаемых пользователями файлов. Одной из главных задач разработчиков является сокращение времени загрузки веб-сайта на стороне клиента. Существует несколько способов достижения этой цели:

1. Уменьшение количества HTTP-запросов

Самый важный совет для ускорения загрузки страницы у посетителей, которые заходят на веб-сайт в первый раз – это уменьшения количества HTTP-запросов. При загрузке страницы большая часть времени уходит на загрузку ее компонентов: картинок, таблиц стилей, скриптов, flash. Уменьшение их количества снижает величину запросов к серверу, необходимых до того, как клиентское приложение сможет отобразить страницу. Это – ключ к созданию быстрых страниц.

Объединений файлов – один из способов повышения производительности веб-приложений. Объединение нескольких скриптов в один скрипт, а несколько таблиц стилей в одну таблицу уменьшает количество запросов на сервер. Объединение файлов наиболее применимо,

когда набор подгружаемых скриптов и CSS отличается от страницы к странице, и этот прием уменьшает общее время загрузки.

CSS-спрайты (*от англ. CSS-sprite*) являются предпочтительным методом сокращения количества запросов на сервер. CSS-спрайты представляют собой несколько различных картинок объединенных в одно целое изображение. Для отображения нужного участка картинки используются CSS-свойства `background-image` и `background-position`.

2. Использование CDN

CDN (*Content Delivery Network*) — это множество веб-серверов, разнесенных географически для достижения максимальной скорости отдачи контента клиенту. Сервер, который непосредственно будет отдавать контент пользователю, выбирается на основании некоторых показателей. Например, выбирается сервер с наименьшим временем отклика.

На скорость загрузки страницы сильно влияет и то, насколько далеко пользователь находится от вашего сервера. Размещение вашего контента между несколькими серверами, разнесенными географически, сделает загрузку сайта быстрее с точки зрения пользователя.

3. Сжатие компонентов страницы

Время, необходимое для пересылки через сеть HTTP-запроса и ответа на него, может быть сильно уменьшено решениями разработчиков фронтэнда (*от англ. front-end - часть программной системы, которая непосредственно взаимодействует с пользователем*). Сжатие помогает уменьшить время загрузки HTTP-ответа, уменьшая его объем.

Начиная с версии протокола HTTP/1.1, веб-клиенты указывают, какие типы сжатия они поддерживают, устанавливая заголовок `Accept-Encoding` в HTTP-запросе.

Если веб-сервер видит такой заголовок в запросе, он может применить сжатие ответа одним из методов, перечисленных клиентом. При выдаче ответа посредством заголовка `Content-Encoding` сервер уведомляет клиента о том, каким методом сжимался ответ.

На данный момент `gzip` является наиболее популярным и эффективным алгоритмом сжатия. Он был разработан проектом GNU и стандартизован в рамках RFC 1952. Единственный алгоритм, который вы можете встретить где-нибудь, кроме `gzip` — это `deflate`, однако он не так эффективен и менее популярен.

В среднем сжатие `gzip` уменьшает размер HTTP-ответа на 70%. Приблизительно 90% используемых сегодня браузеров указывают, что они поддерживают сжатие `gzip`.

Сервер определяет, какие данные нужно сжимать, основываясь на типе файла, но обычно он сильно ограничен в этом выборе. Большинство сайтов сжимают свой HTML. Стоит сжимать скрипты и CSS, а также весь контент, который отдается клиенту текстом (в том числе XML и JSON). Не стоит сжимать картинки и PDF-файлы, так как они уже сжаты. Попытка сжать их не только отнимет процессорное время, но даже может увеличить размер такого файла.

4: Перенос CSS в начале страницы

Спецификация HTML4 устанавливает, что таблицы стилей должны быть включены в HEAD-секцию документа. Разработчики, которые заботятся о производительности своей страницы, всегда хотят, чтобы она могла быть обработана постепенно так как более выгодной является ситуация, когда браузер имеет возможность отобразить любой контент сразу же, как он у него появится. Это особенно важно для страниц, на которых много контента и для пользователей с медленным подключением. Когда браузер загружает страницу постепенно – сначала заголовок, потом навигацию, логотип и так далее – все это служит отличным индикатором загрузки для пользователя, который ожидает страницу. Также это улучшает общее впечатление.

Размещение CSS в конце страницы не позволяет начать постепенную обработку многим браузерам, в числе которых Internet Explorer. Браузер не начинает отображать страницу, чтобы не пришлось перерисовывать элементы, у которых во время загрузки изменится стиль. Браузер Firefox начинает сразу отрисовывать страницу, в процессе загрузки, возможно, перерисовывая некоторые элементы, но это является причиной появления нестилизованного контента

5: Перенос скриптов в конец страницы

Скрипты (внешние .js-файлы): скрипты следует переносить в самый низ страницы, как можно ближе к концу. Делая так, мы позволяем браузеру обрабатывать страницу постепенно и одновременно распараллеливаем загрузку, так как постепенная обработка страницы не начинается для всего контента *ниже* скрипта.

Но в некоторых ситуациях совсем не просто перенести скрипты в конец страницы. Например, если скрипт использует `document.write` для вставки части контента на страницу, такой скрипт не получится перенести вниз.

6: Перенос скриптов и CSS во внешние файлы

Использование подключаемых файлов на практике обычно дает более быстрые страницы, т.к. браузеры кешируют файлы скриптов и CSS. Код javascript и CSS, который встраивается в HTML, загружается каждый раз, когда загружается сам HTML-документ. Это уменьшает

количество необходимых HTTP-запросов, но увеличивает объем HTML. С другой стороны, если скрипты и таблицы стилей находятся в отдельных файлах, скэшированных браузером, размер HTML уменьшается, не увеличивая при этом количество HTTP-запросов.

В таком случае ключевым фактором является частота, с которой кешируются внешние .js- и .css-файлы относительно количества запросов самого HTML-документа. И хотя этот фактор очень сложно посчитать, его можно приблизительно оценить различными способами. Если пользователи сайта во время одного посещения загружают страницу несколько раз или загружают похожие страницы, которые используют один и тот же код – это лучший случай, чтобы получить все преимущества от вынесения кода в отдельные файлы.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. F. Welterlin, D. Ruspini “Presentation Layer Performance Tuning”, - URL: <http://www.scribd.com/doc/9586330/Presentation-Layer-Performance-Tuning>.
2. S. Souders “High Performance Web Sites: The Importance of Front-End Performance”, - URL: http://developer.yahoo.net/blog/archives/2007/03/high_performanc.html.

**А.А. МИТРОШИН, А.А. НОВИКОВ, Р.А. ТАГАНОВ,
С.В. ЧЕРНЫШЕВ**

Рязанский государственный радиотехнический университет

ВОЗМОЖНОСТЬ ИСПОЛЬЗОВАНИЯ МЕТОДА СОСОМО ПРИ ОЦЕНКЕ СЕБЕСТОИМОСТИ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ АСДО ПО ИСПОЛЬЗОВАНИЮ ГЛОНАСС

Рассматривается возможность использования метода СОСОМО для оценки себестоимости разработки АСДО по использованию ГЛОНАСС.

В настоящее время не существует простого метода определения затрат, необходимых для разработки такого достаточно сложного программного продукта, как система дистанционного обучения, и мало надежды на то, что такой метод в ближайшее время будет разработан. Однако, потребность в оценке себестоимости чрезвычайно велика, поскольку серьезные ошибки в оценке стоимости программного обеспечения могут приводить к серьезным последствиям, начиная с нерационального использования средств, заканчивая провалом проекта.

Как правило, организации, разрабатывающие программное обеспечение, оценивают себестоимость программной продукции, одним из методов [2], приведенных в табл. 1.

Табл. 1. Методы оценки себестоимости программного продукта

| Метод | Описание |
|---------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Алгоритмическое моделирование себестоимости | Метод основан на анализе статистических данных о ранее выполненных проектах. Определяется зависимость себестоимости от какого-либо показателя программного продукта, проводится его оценка для данного проекта и прогнозируются будущие затраты. |
| Экспертная оценка себестоимости | Проводится опрос нескольких экспертов по технологиям разработки программного обеспечения. Каждый из них дает свою оценку себестоимости проекта. Все оценки сравниваются и обсуждаются. Этот процесс продолжается до тех пор, пока не будет достигнуто согласие по окончательному варианту предварительной себестоимости проекта. |
| Оценка по аналогии | Метод используется в том случае, если в области применения разрабатываемого программного продукта уже реализованы аналогичные проекты. Оценка себестоимости текущего проекта оценивается по аналогии с ранее реализованными проектами. |
| Закон Паркинсона | Критерием оценки себестоимости проекта являются человеческие ресурсы, а не целевая оценка самого программного продукта. |
| Назначение цены с целью выиграть контракт | Себестоимость определяются наличием тех средств, которые есть у заказчика. |

Из табл. 1 видно, что алгоритмическое моделирование является наиболее адекватным подходом к определению себестоимости программного обеспечения АСДО. Алгоритмическую модель стоимости можно построить с помощью анализа затрат и параметров уже реализованных проектов, в том числе и аналогичных.

В большинстве алгоритмических моделей выражения для оценки себестоимости имеют экспоненциальный вид. Это связано с тем, что с усложнением проекта появляются дополнительные расходы, связанные с ростом затрат на коммуникации (в том числе и расходы на ко-

мандировки), усложнением управления конфигурацией, увеличением объема работ по сборке системы и т.д. Оценки затрат также могут умножаться на коэффициенты, учитывающие свойства разрабатываемого программного продукта, платформу разработки, технологию создания программного продукта и квалификацию разработчиков и т.п.

В общем случае выражение для вычисления алгоритмической оценки себестоимости может быть определено следующим образом:

$$\text{Затраты} = A * \text{Размер}^B * M,$$

где A – постоянный коэффициент, зависящий от организации – исполнителя проекта и типа разрабатываемого программного обеспечения; Размер может соотноситься либо с размером программного кода, либо с функциональной оценкой, выраженной в количестве объектных или функциональных точек; B отражает объем работ, требующихся для реализации больших проектов (варьируется в пределах от 1 до 1,5); M отображает характеристики различных этапов разработки, а также характеристики создаваемого проекта.

Одной из самых распространенных алгоритмических моделей для оценки стоимости программного продукта является модель СОСОМО (СОnstructive СОst MOdel – конструктивная стоимостная модель). Эта модель основана на опыте выполнения многих программных проектов. Она популярна по следующим причинам [3]:

- фактические данные собираются в соответствии со многими реальными проектами;
- позволяет добавлять уникальные факторы для корректировки экономических характеристик;
- хорошо подходит для проектов, между которыми нет существенных различий относительно размеров;
- модель прошла достаточно длинный путь развития. Первая версия модели была опубликована в 1981 году (эта версия известна в настоящее время как СОСОМО 81). Версия модели СОСОМО 2 опубликована в 1995 году.

Модель СОСОМО 81 имеет трехуровневую структуру [1], где уровни определяют сложность анализа себестоимости.

На первом (базовом) уровне проводится начальная грубая оценка, на втором уровне эта оценка уточняется путем применения различных множителей, учитывающих особенности программного проекта и технологии разработки программного обеспечения исполнителя, третий уровень дает возможность рассчитывать себестоимость для разных стадий проекта. В табл. 2 приведены основные расчетные формулы для проектов с различной степенью сложности. В приводимых выражениях:

- PM (Person-Months) – человеко-месяцы, необходимые для реализации проекта;

- KDSI (thousand (Kilo-) of Delivered Source Instructions) – количество инструкций (в тысячах) в конечной программе.

Моделью COCOMO 81 предусмотрена разработка программного обеспечения в соответствии с каскадной [2] (водопадной) моделью, причем предполагается, что большая часть системы разрабатывается с самого начала.

Табл. 2 Расчетные выражения модели COCOMO 81

| Сложность проекта | Выражение | Описание |
|---------------------------|----------------------------------|-----------------------------------------------------------------------------------------------------------|
| Простой проект | $PM = 2,4(KDSI)^{1,05} \times M$ | Простые проекты для небольших команд разработчиков |
| Проект средней сложности | $PM = 3,0(KDSI)^{1,12} \times M$ | Более сложные проекты, в которых исполнитель может испытывать недостаток опыта и знаний |
| Проект встроенной системы | $PM = 3,6(KDSI)^{1,20} \times M$ | Сложные проекты, в которых программный продукт является частью комплекса программных и аппаратных средств |

Модель применима и при производстве программного обеспечения путем компоновки повторно используемых компонентов.

Модель COCOMO 2 допускает использование разнообразных подходов к процессу разработки программных продуктов: прототипирование, сборку систем из отдельных компонентов, использование языков программирования четвертого поколения и т.д. В COCOMO 2 уровни модели отображают не только возрастающую сложность определения себестоимости разработки программного обеспечения, но и учитывают этапы работы над программой. Это позволяет провести предварительную оценку себестоимости уже на ранних стадиях выполнения проекта с последующей ее детализацией после определения архитектуры системы.

Модель COCOMO 2 охватывает три следующих уровня:

- уровень предварительного прототипирования. Для определения затрат осуществляется оценка размера системы на основе объектных точек прототипа с помощью формулы «размер-производительность»;

- уровень предварительного проектирования. Этот уровень предусматривает окончание работы над системными требованиями и, возможно, над начальным проектом архитектуры системы. Оценка затрат на этом уровне основана на функциональных точках, которые затем пересчитываются в количество строк программы;

- постархитектурный уровень. После разработки архитектуры системы существует возможность относительно точно оценить размер будущей программы. Оценка на этом уровне будет включать большое количество множителей, которые должны отражать возможности персонала, а также характеристики создаваемого программного продукта в целом.

Уровень предварительного прототипирования введен в модель для оценки затрат на прототипирование проектов, а также тех проектов, в которых программное обеспечение разрабатывается путем сборки из уже существующих компонентов. Здесь оценка основана на подсчете взвешенных объектных точек, деленных на стандартное значение производительности разработки. Производительность зависит от опыта и способностей разработчиков, а также от возможностей средств, используемых для поддержки процесса разработки. В табл. 3 показаны различные уровни производительности, которые были предложены разработчиками модели COCOMO 2.

Табл. 3. Производительность, выраженная в объектных точках (PROD)

| | | | | | |
|---------------------------------------------------------|--------------|--------|---------|---------|---------------|
| Опыт и возможности программиста | Очень низкие | Низкие | Средние | Высокие | Очень высокие |
| Уровень и возможности CASE-средств | Очень низкие | Низкие | Средние | Высокие | Очень высокие |
| Производительность (количество объектных точек в месяц) | 4 | 7 | 13 | 25 | 50 |

На этом уровне достаточно распространено повторное использование ранее разработанных компонентов, поэтому прогнозируемое количество объектных точек должно учитывать процентное соотношение повторно используемых компонентов (*% многократного использования*). Таким образом, формула для предварительного определения объема работ выглядит следующим образом:

$$PM = (NOP \times (1 - \% \text{ многократного использования} / 100)) / PROD.$$

Здесь PM – это затраты, выраженные в человеко-месяцах, NOP – количество объектных точек, а PROD – производительность (таблица 3).

На уровне предварительного проектирования оценка основана на стандартной форме алгоритмических моделей

$$\text{Затраты} = A * \text{Размер}^B * M.$$

Для оценки на этом уровне часто используется $A=2,5$.

Размер системы выражается в KSLOC (количество строк программного кода в тысячах). Он определяется путем подсчета количества функциональных точек в программе и перевода их в KSLOC с помощью стандартных таблиц, которые определяют соотношение размера программы с функциональными точками для различных языков программирования. Эта оценка достаточно хорошо применима к кодам, написанным вручную и значительно хуже к коду, сгенерированному с помощью визуальных построителей или повторно используемым компонентам. Поскольку при разработке АСДО по использованию ГЛЮНАСС доля программного кода, написанного вручную значительна, то оценка может быть достаточно объективной.

Показатель степени В отражает затраты, которые увеличиваются по мере увеличения размера проекта. Это не постоянная величина, её значение может изменяться от 1,1 до 1,24, что определяется тем, насколько новаторским является данный проект, от гибкости процесса разработки программного обеспечения, от применяемых процессов управления рисками, сплоченности команды программистов и уровня управления в организации-разработчике.

Множитель М является произведением семи показателей, характеризующих проект и процесс создания программного продукта, а именно: надежность и уровень сложности разрабатываемой системы (*RCPX*), повторное использование компонентов (*RUSE*), сложность платформы разработки (*PDIF*), возможности персонала (*PERS*), опыт персонала (*PREX*), график работы (*SCED*) и средства поддержки процесса разработки (*FCIL*). Это позволяет произвести оценивание по шестибальной шкале, где число 1 будет соответствовать самым малым

значениям этих показателей, а число 6 – самым высоким. Таким образом, оценка затрат вычисляется по следующему выражению:

$$PM = A * \text{Размер}^B * M + PM_m,$$

где

$$M = PERS \times RCPX \times RUSE \times PDIF \times PREX \times FCIL \times SCED.$$

Последнее слагаемое (PM_m) обозначает фактор, используемый в случаях, когда значительная часть кода генерируется автоматически. При этом часть кода требуется вводить вручную, но уровень производительности будет выше, чем при полностью ручном вводе. Затраты PM_m оцениваются по следующей формуле:

$$PM_m = (ASLOC \times (AT / 100)) / ATPROD,$$

где *ASLOC* – количество строк кода, произведенных автоматическим способом, *ATPROD* – уровень производительности автоматической генерации кода. Следует заметить, что требуется выполнение определенных работ для согласования сгенерированного кода с остальной частью программной системы. Объем этих работ зависит от процента автоматически сгенерированного кода во всей системе (коэффициент *AT*). Фактически производительность зависит от количества созданных программных модулей. Чем меньше объем сгенерированного кода, тем больший объем работ необходим для интеграции его с другим кодом системы.

Для получения оценки на постархитектурном уровне используется такое же выражение, как и для оценки на уровне предварительного проектирования. На этом уровне оценка будет более точной, для оценки будут использоваться не семь, а семнадцать показателей.

При оценке количества программного кода учитываются следующие важные факторы:

- возможность изменения системных требований. Следует учитывать повторные работы, которые необходимо выполнить вследствие изменения системных требований. Оценка этих работ выражается в количестве строк программного кода, которое необходимо изменить, и затем прибавляется к предварительной оценке размера системы;

- степень повторного использования компонентов. Если степень повторного использования компонентов значительна, в оценку количества строк разрабатываемого кода необходимо внести поправки. Однако, расходы на повторное использование компонентов не всегда линейно зависят от размера этих компонентов так как требуют затрат на их подбор и на то, чтобы разобраться с их возможностями и интерфейсами. Также могут потребоваться затраты на внесение изменений в компоненты.

Оценка затрат на повторное использование компонентов в модели COCOMO 2 рассчитывается в соответствии с выражением:

$$ESLOC = ASLOC \times (AA + SU + 0.4 \times DM + 0.3 \times CM + 0.3 \times IM) / 100,$$

где $ESLOC$ – количество строк нового кода, $ASLOC$ – количество строк повторно используемого кода, требующего изменений, DM – процент изменений в архитектуре системы, CM – процент измененного кода, IM – процент затрат на интеграцию повторно используемого программного обеспечения. Величина SU зависит от затрат на адаптацию повторно используемых программных компонент и колеблется в пределах от 50 (для сложного и плохо структурированного кода) до 10 (для грамотно написанного объектно-ориентированного кода). Величина AA отражает затраты на начальную оценку возможности повторного использования компонентов. Его значение может меняться от 0 до 8.

Показатель степени в модели COCOMO 81 имеет три возможных значения, которые относятся к проектам различной сложности. С возрастанием уровня сложности проекта увеличивается значимость размера системы. Однако отрицательный эффект размера системы можно нивелировать с помощью организационных мероприятий, что учтено в модели COCOMO 2. Здесь показатель степени рассчитывается с учетом пяти показателей, которые приведены в табл. 4. Они отсчитываются по шестибальной шкале от низшего (5 баллов) до высшего (0 баллов) уровня. Значения показателей суммируются, сумма делится на 100, результат прибавляется к числу 1,01, после чего получается значение показателя степени.

Табл. 4. Показатели, используемые при подсчете показателя степени в модели COCOMO 2

| Показатель | Пояснение |
|------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Новизна проекта | Отражает предыдущий опыт организации в реализации проектов данного типа. Очень низкий уровень этого показателя означает отсутствие опыта, наивысший уровень указывает на компетентность организации-разработчика в данной области программного обеспечения |
| Гибкость процесса разработки | Отражает возможность изменения процесса разработки программного обеспечения. Очень низкий уровень этого показателя означает, что процесс определен заказчиком заранее, наивысший – заказчик определил лишь общие задания без указания конкретной технологии процесса разработки программной системы |

| Показатель | Пояснение |
|--------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Анализ архитектуры системы и рисков | Отображает степень детализации анализа рисков, основанного на анализе архитектуры системы. Очень низкий уровень данного показателя соответствует поверхностному анализу рисков, наивысший уровень означает, что был проведен тщательный и полный анализ всевозможных рисков |
| Сплоченность команды | Отражает степень сплоченности команды разработчиков организации-заказчика, то есть ее способность работать совместно. Очень низкий уровень этого показателя означает, что взаимоотношения в команде сложные, а наивысший – что команда сплоченная и эффективная в работе, не имеет проблем во взаимоотношениях |
| Уровень развития процесса разработки | Отображает уровень развития процесса создания программного обеспечения в организации-разработчике. Оценка этого показателя основывается на вопроснике модели СММ, описанной ниже |

Проектные характеристики, используемые для уточнения предварительной оценки затрат на постархитектурном уровне, разбиваются на 4 группы.

1) Характеристики программного продукта, которые определяются системными требованиями.

2) Характеристики аппаратных средств, представляющие собой ограничения, накладываемые на разрабатываемое программное обеспечение выбранной платформой вычислительных средств.

3) Характеристики персонала, которые учитывают опыт и возможности специалистов, работающих над проектом.

4) Характеристики проекта, учитывающие определенные параметры и показатели проекта разработки программного продукта.

Каждая из описанных групп, включает параметры, представленные в табл. 5.

Табл. 5. Проектные характеристики, формирующие стоимость проекта

| Характеристики программного продукта | |
|--------------------------------------|---------------------------------|
| RELY | Требуемая надежность системы |
| CPLX | Сложность системных модулей |
| DOCU | Объем необходимой документации |
| DATA | Размер используемой базы данных |

| | |
|-----------------------------------|-----------------------------------------------------------------------|
| RUSE | Процент повторного использования компонентов |
| Характеристики аппаратных средств | |
| TIME | Показатели, ограничивающие время выполнения |
| PVOL | Возможность изменения платформы разработки |
| STOR | Ограничение объема памяти |
| Характеристики персонала | |
| ACAP | Способности лиц, выполняющих анализ проекта |
| PCON | Сплоченность команды разработчиков |
| PEXP | Опыт программирования в данной области программного обеспечения |
| PCAP | Способности программистов |
| AEXP | Опыт аналитика проекта в данной области программного обеспечения |
| LTEX | Опыт применения данного языка программирования и средств разработки |
| Характеристики проекта | |
| TOOL | Использование вспомогательных программных средств |
| SCED | Уплотнение графика работ |
| SITE | Количество работ, выполняемых в разных местах и качество коммуникаций |

В табл. 6 указано, каким образом эти характеристики влияют на предварительную оценку затрат. Если использовать показатель степени равный 1,17, и предположить, что основными характеристиками, формирующими стоимость проекта, являются требуемая надежность системы (RELY), сложность системных модулей (CPLX), ограничение объема памяти (STOR), использование вспомогательных программных средств (TOOL) и уплотнение графика работ (SCED), а остальные параметры не являются значимыми и, следовательно, равны 1, то получим расчет оценки затрат, представленный в табл. 6. Отметим, что приводимые данные предназначены для демонстрации влияния некоторых характеристик проекта на оценку его себестоимости и никак не связаны с разработкой АСДО по изучению ГЛОНАСС.

Табл. 6. Расчет оценки затрат

| | |
|--------------------------------------------------------------------------------------------------|------------------------------------------------------------|
| Значение показателя степени | 1,17 |
| Размер системы (с учетом повторного использования компонентов и возможного изменения требований) | 128 000 (DSI – количество инструкций в конечной программе) |
| Начальная оценка по модели | 730 человеко-месяцев |

| | |
|-------------------------------------------------|--------------------------------|
| СОСОМО без учета проектных характеристик | |
| Надежность системы | Очень высокая – множитель 1,39 |
| Сложность системных модулей | Очень высокая – множитель 1,3 |
| Ограничения объема памяти | Высокое – множитель 1,21 |
| Использование вспомогательных средств | Низкое – множитель 1,12 |
| График работ | Ускоренный – множитель 1,29 |
| Уточненная оценка по модели СОСОМО | 2306 человеко-месяцев |
| Надежность системы | Очень низкая – множитель 0,75 |
| Сложность системных модулей | Очень низкая – множитель 0,75 |
| Ограничение объема памяти | Нет – множитель 1,0 |
| Использование вспомогательных средств | Очень высокое – множитель 0,72 |
| График работы | Нормальный – множитель 1,0 |
| Уточненная оценка по модели СОСОМО | 295 человеко-месяцев |

В этом примере ключевым характеристикам присвоены максимальные и минимальные значения для того, чтобы показать, каким образом они влияют на оценку затрат программного продукта.

Таким образом алгоритмические методы оценки стоимости программного обеспечения, в частности СОСОМО 2, могут использоваться для оценки себестоимости разработки АСДО по использованию ГЛОНАСС.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Макконел С. Сколько стоит программный проект. – М.: «Русская редакция», СПб.: Питер, 2007.
2. Соммервил И. Инженерия программного обеспечения, 6-е издание. – М.: Издательский дом «Вильямс», 2002.
3. Липаев В.В. Экономика производства сложных программных продуктов. – М., СИНТЕГ, 2008.

Н.А. ПОДГОРНОВА

Рязанский государственный радиотехнический университет

ПРИМЕНЕНИЕ ВЕЙВЛЕТ-АНАЛИЗА И ВЕЙВЛЕТ-ПРЕОБРАЗОВАНИЯ ДЛЯ ИССЛЕДОВАНИЯ И ОБРАБОТКИ ВРЕМЕННЫХ РЯДОВ ДЕНЕЖНЫХ ПОТОКОВ ПРЕДПРИЯТИЯ

Рассматриваются применение *Wavelet* - преобразования для анализа состояния и нестационарных процессов, получения информации о возможности прогнозирования финансовых данных, фильтрации и подавлением шумов, предварительной обработки данных и прогнозирования ситуации, отыскивания признаков фрактальности информации.

Управление денежными потоками предприятия, являющейся важной составной частью общей системы управления его финансовой деятельностью. Повышению эффективности процесса управления денежными средствами в значительной степени способствует анализ, планирование и прогнозирование денежных потоков и последующая их оценка для принятия своевременных управляющих решения на российских предприятиях, что является особенно актуальным в условиях кризиса.

С помощью пакета *Statistica 6.1* осуществлен анализ денежных потоков предприятия ОАО «АВС» классическими методами статистики. Из полученных расчетов статистических показателей следует, что временные ряды денежных потоков предприятия ОАО «АВС» не подчиняются Гауссовому распределению и описываются распределениями с «толстыми» хвостами и высокими пиками, что и показано на графике рис. 1 и 2 распределения ежедневных потоков и аппроксимирующих их нормальное распределение.

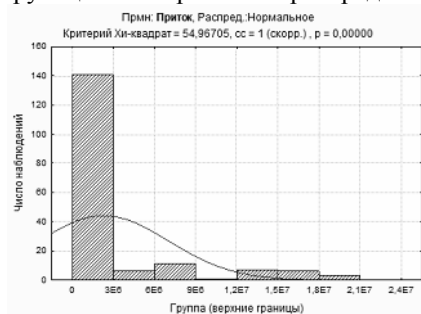


Рис. 1. График функции нормального распределения временного ряда притока денежных средств предприятия ОАО «АВС»

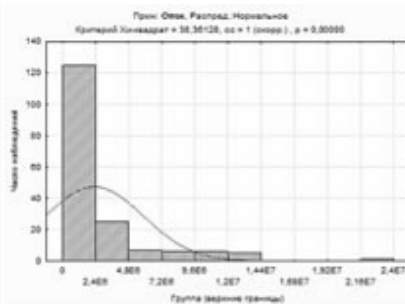


Рис. 2. График функции нормального распределения временного ряда оттока денежных средств предприятия ОАО «АВС»

Из анализа временных рядов денежных потоков предприятия следует, что временные ряды денежных средств предприятия являются нестационарными. Временные ряды денежных потоков сильно зашумлены и краткосрочные, слабовыраженный тренд может восприниматься как случайные колебания. Необходимо провести анализ финансовых данных и определить такие факторы, как периодичность, тренд, локальные особенности.

Для получения предпрогнозной оценки и возможности прогнозирования исследуемого временного ряда мы считаем, что можно использовать новый способ обработки и исследования сигналов – вейвлет-анализ, который широко применяется для анализа нестационарных (во времени) и неоднородных (в пространстве) сигналов. Известны трудности, встречающиеся при обработке коротких высокочастотных сигналов или сигналов с локализованными частотами. Вейвлет-преобразование оказывается очень удобным инструментом для адекватной расшифровки таких данных, поскольку элементы его базиса хорошо локализованы и обладают подвижным частотно-временным окном.

Анализ данных с использованием непрерывного вейвлет-преобразования является удобным, надежным и мощным инструментом исследования денежных потоков и позволяет представить результаты в очень наглядном виде, удобном для изучения и интерпретации.

Построим график сигнала, в качестве которого представлены временные ряды денежного потока предприятия ОАО «АВС» ($N=175$ – количество наблюдений временного ряда ежедневных изменений потоков денежных средств) и вейвлет-спектрограмму сигнала – графическое изображение вейвлет-коэффициентов сигнала (рис. 3 и 4). На спектрограмме представлены периоды циклов.

Для реализации непрерывного вейвлет – преобразования используем прикладную программу *Matlab 7*. Осуществим вейвлет – преобразование с гауссовым вейвлетом (*gaus1*), а спектрограмму построим с помощью функции *cwt*.

На рис. 3 - 4 изображены вейвлетограммы непрерывного вейвлет – преобразование с использованием вейвлета Гаусса порядок 1 (центральная частота $Fr=0,2$). Вейвлеты типа Гауссова относятся к грубым вейвлетам. Низкие частоты (крупный масштаб) для получения глобальной информации о сигнале (обычно охватывает весь сигнал). Высокие частоты (мелкий масштаб) – для детальной информации, скрытых, локальные особенности сигналов (обычно продолжается не очень долго). В нижней части спектрограмм отчетливо видны частые изме-

нения яркости, указывающие на наличие периодических высокочастотных компонентов.

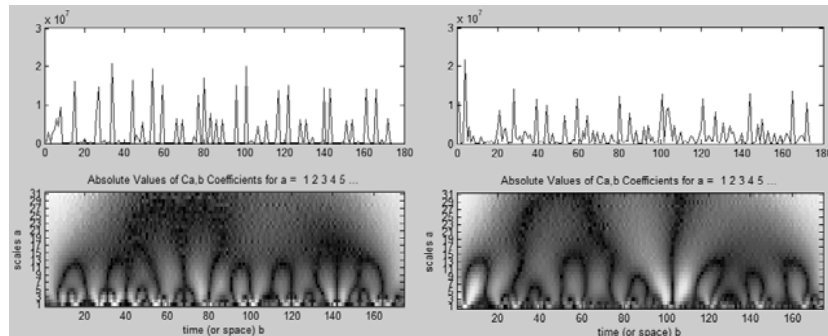


Рис. 3. График и спектрограмма сигнала ежедневного притока денежных средств предприятия ОАО «ABC» ($a=[1:1:32]$, $t=[1:175]$)

Рис. 4. График и спектрограмма сигнала ежедневного притока денежных средств предприятия ОАО «ABC» ($a=[1:1:32]$, $t=[1:175]$)

В верхней части заметны менее частые изменения яркости, соответствующие низкочастотным компонентам. На графиках представлены большое количество циклов с маленькими периодами, которые необходимо учитывать. На графике денежного потока видны вертикальные черные кривые, демонстрирующие моменты изменения поведения и структуры денежного оттока. Эти периоды («пики») притока денежных средств представляют собой области непредсказуемости. Отток денежных средств является планируемым и более предсказуемым. В моменты выбросов и скачков оказывают случайные обстоятельства и внешние факторы. Области между «пиками» являются хорошо прогнозируемыми областями.

Постояим спектрограммы сигналов денежных потоков (рис. 5 – 6), синтезированные вейвлетом Добеши ($db1$, центральная частота $Fr = 0.9961$). Вейвлет Добеши относится к ортогональным вейвлетам с компактным носителем (используется для непрерывного и дискретного преобразования с применением быстрого вейвлет-преобразования, обладает уникальным свойством – возможность точной реконструкции сигналов произвольного вида).

На графике (рис. 5 - 6) отчетливо видны локальные особенности и пики сигналов. Локальным особенностям (нарушениям гладкости) отвечают вертикальные полосы, выходящие из точки, где находится особенность. Пикам сигналов соответствует сгущение светлых областей вейвлет-спектрограмм, а впадинам – сгущение темных областей. Чем резче выражена особенность сигнала, тем сильнее она выделяется на спектрограмме.

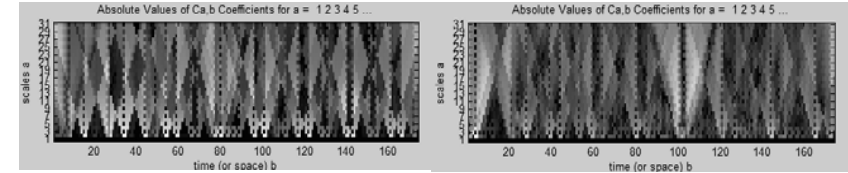


Рис. 5. Спектрограмма преобразования сигнала притока ДС вейвлетом Добеши ($a=[1:1:32]$, $t=[1:175]$)

Рис. 6. Спектрограмма преобразования сигнала оттока ДС вейвлетом Добеши ($a=[1:1:32]$, $t=[1:175]$)

На рис. 7 - 8 представлены трехмерные спектры сигналов денежных потоков, синтезированные гауссовым вейвлетом ($Gaus1$, центральная частота $Fr = 0.2000$, где X – время, Y – частота, Z – амплитуда гармоники с данной частотой в данный момент времени).

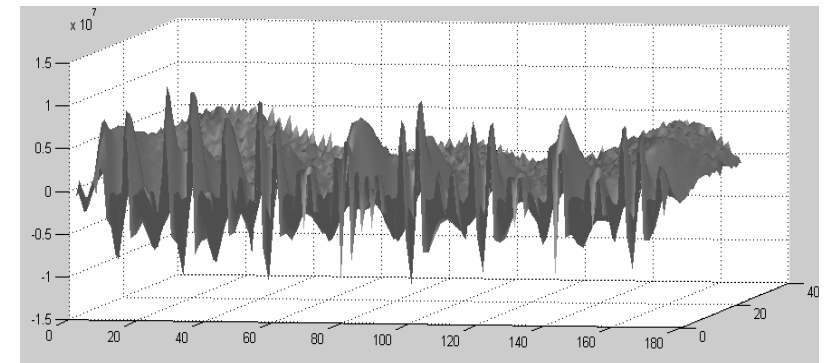


Рис. 7. Спектрограмма преобразования сигнала притока ДС вейвлетом Гаусса в трехмерном измерении.

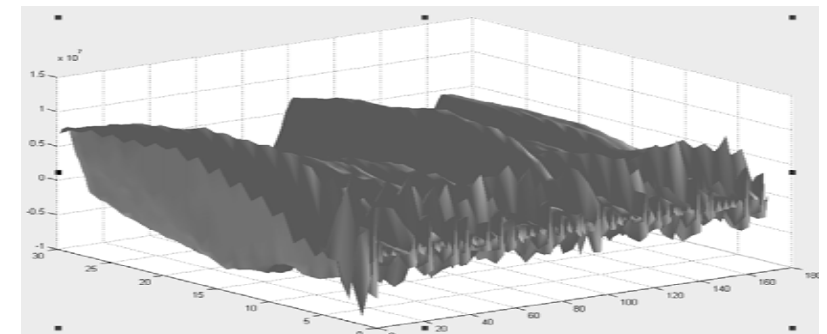


Рис. 8. Спектрограмма преобразования сигнала оттока ДС вейвлетом Гаусса в трехмерном измерении.

Непрерывное wavelet-преобразование использовали для анализа переходных процессов, обнаружения резких изменений в сигнале и исследовании нестационарностей. Применение wavelet-преобразования дали хорошие результаты т.к. компоненты исследуемого сигнала потоков денежных средств предприятия с высокой частотой имеют небольшую длительность, а низкочастотные компоненты – достаточно большую.

Непрерывные вейвлеты дают несколько более наглядное представление результатов анализа в виде поверхностей вейвлет-коэффициентов по непрерывным переменным. Преобразование вейвлетом Гаусса и Добеши четко показывает их пространственную и частотную локализацию, в то время как спектр Фурье дает только частотную локализацию. Однако базисы на основе непрерывных вейвлетов, как правило, не являются строго ортонормированными, поскольку элементы базиса бесконечно дифференцируемы и экспоненциально спадают на бесконечности. У дискретных вейвлетов эти проблемы легко снимаются, что обеспечивает более точную реконструкцию сигналов.

С помощью пакета расширения *Wavelet Toolbox* системы *Matlab7* выполняем декомпозицию сигнала потоков денежных средств предприятия до уровня 2 с использованием вейвлета *Добеши db3*

На графиках представлена декомпозиция сигнала в виде исходного сигнала (s) и его грубое (аппроксимация) - детальное приближение, представленные коэффициентами 2-х уровней (a_2, d_1, d_2)

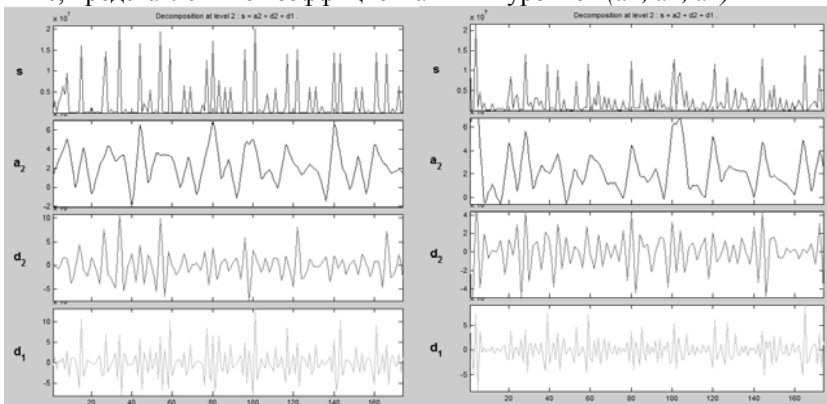


Рис. 9. Графики аппроксимации верхнего 2-го уровня разложения и детализирующих компонент сигнала денежного притока

Рис. 10. Графики аппроксимации верхнего 2-го уровня разложения и детализирующих компонент сигнала денежного оттока

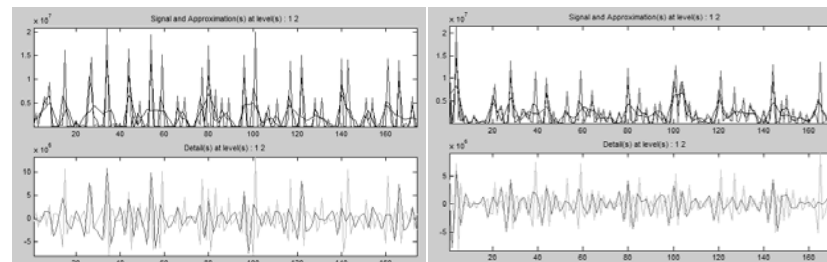


Рис. 11. Графики сигнала и аппроксимации (в верхнем окне), детали компонент сигнала денежного притока (в нижнем окне)

Рис. 12. Графики сигнала и аппроксимации (в верхнем окне), детали компонент сигнала денежного оттока (в нижнем окне)

Исследуемый временной ряд денежных потоков хаотичен и большое значение имеет выбор конкретного вида и типа вейвлетов для изучения высших частот, удаление шума и сглаживания данных. Для этого необходимо задать глубину разложения и порядок вейвлета. Выбор используемого вейвлета и глубины разложения, в общем случае, зависит от свойств конкретного сигнала. Порядок вейвлета определяет гладкость восстановленного ряда данных. Более гладкие вейвлеты (7-го порядка), создают более гладкую аппроксимацию сигнала - «выбросы» будут сглажены, и наоборот – «короткие» (2-го порядка), вейвлеты лучше отслеживают пики аппроксимируемой функции – ярче будут выражены «выбросы». Глубина разложения влияет на масштаб отсеиваемых деталей: при увеличении глубины разложения модель (7-9 уровней разложения) вычитает шум все большего уровня (более «крупные» детали в исходных данных будут отброшены), пока не наступит «переукрупнение» масштаба деталей и преобразование начнет искажать форму исходного сигнала. При дальнейшем увеличении глубины разложения преобразование начинает формировать сглаженную версию исходного сигнала («обрезаются» резкие выбросы), т.е. отфильтровывается не только шум, но и некоторые локальные особенности (выбросы) исходного сигнала.

Произведем очищению от шума временные ряды потоков денежных средств с использованием вейвлет-преобразования, короткого вейвлета Добеши 3-го порядка, 2 уровня разложения (пакет *Wavelet Toolbox* системы *Matlab 7*) (рис. 13 – 14) для дальнейшего исследования и прогнозирования.

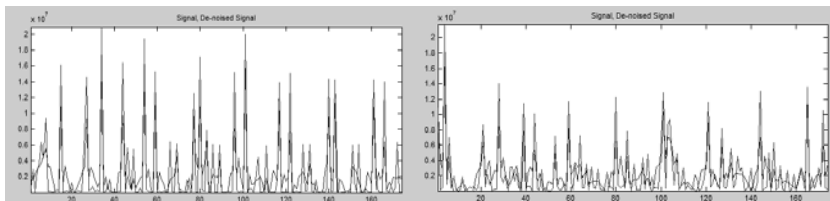


Рис. 13. Исходный временной ряд притока денежных средств предприятия и после очищении от шума

Рис. 14. Исходный временной ряд оттока денежных средств предприятия и после очищения от шума

Вейвлеты используем как механизм обработки и исследования данных. Вейвлет-преобразование дает наиболее наглядную и информативную картину результатов эксперимента, позволяет очистить исходные данные от шумов и случайных искажений, подметить некоторые особенности данных и направление их дальнейшей обработки и анализа. Вейвлеты хорошо подходят для анализа нестационарных сигналов, характерные для финансовых данных. Вейвлет-анализ для предпрогнозного исследования и анализа временного ряда денежных потоков предприятия. Вейвлет-анализ позволяет лучше локализовать процесс во времени по сравнению с фрактальным анализом. Большие трудности при обучении нейросетей (или настройке других механизмов анализа данных) создает сильная зашумленность данных или наличие большого числа непредсказуемых случаев (случайные выбросы, пропуски, нелинейные искажения и т.п.). Такие помехи способны скрывать характерные особенности данных или выдавать себя за них и могут сильно ухудшить результаты обучения. Поэтому необходимо очистить данные перед их анализом и прогнозом. Благодаря наличию быстрых и эффективных алгоритмов реализации, вейвлеты представляются весьма удобным и перспективным механизмом очистки и предварительной обработки данных для использования их в статистических и экономических программах и комплексах, системах искусственного интеллекта. Обработанные и исследованные на основе инструментария вейвлетов временные ряды денежных потоков позволяют значительно повысить обоснованность управленческих решений по формированию резерва остатков денежных средств на предприятии.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Добеши Я. Десять лекций по вейвлетам. М.: РХД, 2001.
2. Дьяконов В. MATLAB/ Обработка сигналов и изображений. Специальный справочник. – СПб: Питер, 2002. – 608 с.
3. Смоленцев Н.К. Основы теории вейвлетов. Вейвлеты в MATLAB – ДМК Пресс, 2005.-304 с.

М.С. РОЗУМЕЕНКО

Рязанский государственный радиотехнический университет

СЕМАНТИЧЕСКИЕ ОШИБКИ В SQL ЗАПРОСАХ И ПРИНЦИПЫ ИХ ВОЗНИКНОВЕНИЯ

Рассматриваются основные семантические ошибки в SQL запросах, причины их возникновения, а так же некоторые предложения по оптимизации некорректного SQL кода.

Ошибки в SQL запросах можно разделить на синтаксические и семантические. Синтаксическая ошибка означает, что введенный символ строки не является допустимым. В этом случае, любая СУБД выдаст сообщение об ошибке, поскольку не сможет выполнить такой запрос. Таким образом, синтаксические ошибки легко обнаружить и исправить.

Семантическая ошибка означает, что синтаксически корректный SQL запрос не выдает или не всегда выдает желаемый результат, и, следовательно, неприменим для данной задачи.

Причины возникновения семантических ошибок можно разделить на два класса:

- когда запрос явно не соответствует заданной логике
- когда нет достаточных доказательств того, что запрос неправильный.

Остановимся подробнее на последнем классе. Рассмотрим в качестве примера следующий запрос:

```
SELECT *
FROM EMPLOYERS
WHERE JOB = 'CLERK' AND JOB = 'MANAGER'
```

Этот синтаксически верный запрос выполняется, например, в СУБД Oracle8i без всякого предупреждения. Без семантической оптимизации данный запрос будет выполняться следующим образом — будет последовательно извлекаться каждая запись в таблице 'EMPLOYERS' и проверяться на выполнение условия отбора. Результатом выполнения запроса будет пустое множество записей. Поскольку пустой результат не несет никакого смысла, можно констатировать, что этот запрос является неправильным, невзирая на изначальную цель выполнения данного запроса. С учетом внутренней семантической оптимизации в ответ на запрос без последовательного перебора всех записей сразу будет выдано пустое множество записей.

Хотя задачи такого типа неразрешимы в целом, во многих случаях на практике могут быть решены с помощью относительно простых

алгоритмов. Инструмент для поиска семантических ошибок в SQL-запросах будет полезен не только для обучения, но и в разработке приложений программного обеспечения. По крайней мере, своевременное сообщение об ошибке может ускорить процесс отладки.

Ниже представлен перечень наиболее часто встречаемых семантических ошибок, а также общие принципы их возникновения. В статье рассмотрены основные причины, по которым SQL запросы могут считаться подозрительными: избыточность, неэффективная формулировка, нарушения стандартных правил, выборка дубликатов, а также возможные ошибки выполнения.

Конечно, в целом трудно понять, в каком случае синтаксически правильный запрос семантически неверен, если человек не знает, задачу, для которой запрос был написан. Тем не менее, запросы могут рассматриваться как "вероятно, неверные", когда они излишне сложные. Предположим, пользователь пишет запрос Q, и есть эквивалентный запрос Q', который значительно проще и который может быть получен из Q путем исключения некоторых условий.

Существует несколько причин, почему пользователь не написал Q': например, пользователь знал, что логика работы Q' не соответствует поставленной задаче, либо пользователь не знал, что Q и Q' эквивалентны.

Одним из проявлений вариантности языка SQL является эквивалентность выражения некоторых подчиненных и сложных запросов с соединениями. При выполнении запросов с подчиненными запросами для каждого кортежа-записи в исходном наборе внешнего запроса выполняется подчиненный запрос. Иначе говоря, всякий раз при вычислении предиката внешнего запроса вычисляется подчиненный запрос. Поэтому резервом для оптимизации подобных запросов является поиск возможных путей сокращения количества операций за счет эквивалентных преобразований, приводящих к совмещению операций формирования набора кортежей-записей внешнего и внутреннего (подчиненного) запроса.

Таким образом, семантическая оптимизация запросов основывается на слиянии внутреннего представления запроса и ограничений целостности конкретной базы данных до непосредственного выполнения запроса и призвана за счет совместной проверки ограничений целостности и условий запроса сократить количество выполняемых операций.

Ниже представлен список условий, с помощью которых запрос может быть упрощен. В каждом из этих случаев, пользователю должно выводиться предупреждение о возможной ошибке.

1. Избыточные условия в запросе
Error 1: Несовместимые условия
2. Лишнее условия в операторе SELECT
Error 2: Исключение ненужных дубликатов
Error 3: Выходное поле с константой
Error 4: Дублирующееся выходное поле
3. Лишнее условия в операторе FROM
Следующие ошибки возникают в тех случаях, когда объявленные в условии FROM переменные по факту не используются.
Error 5: Неиспользуемые переменные
Error 6: Неиспользуемый JOIN
Error 7: Одинаковые переменные
4. Лишние условия в операторе WHERE
Error 8: Несовместимые подусловия
Error 9: Неиспользуемый оператор сравнения
Error 10: Неиспользуемые аргументы в подзапросах
Error 11: Условия IN / EXISTS могут быть заменены сравнением
5. Лишние условия в функциях агрегации
Error 12: Ненужный DISTINCT
Error 13: Ненужные аргумент для функции COUNT
6. Лишние условия в операторе GROUP BY
Error 14: GROUP BY для одиночных значений
Error 15: GROUP BY для одной группы
Error 16: Лишние атрибуты GROUP BY
7. Лишние условия в операторе HAVING
В операторе HAVING встречаются те же ошибки, что и в операторе WHERE
8. Лишние условия в операторе ORDER BY
Error 17: Лишний оператор ORDER BY
9. Декларативный характер языка SQL («что сделать») приводит к неоднозначности в определении конкретной схемы и конкретного порядка обработки данных (наличие нескольких вариантов «как сделать»). Еще одним сигналом возможной ошибки может стать отклонение от стандартных структур:
Error 18: Отсутствуют условия JOIN
Error 19: Несвязанные EXISTS-подзапросы
Error 20: оператор SELECT не использует выборку подзапроса
Error 21: Условия подзапроса могут быть повышены
Error 22: Сопоставление различных доменов
Error 23: Нехарактерный HAVING (без GROUP BY)
Error 24: Групповой символ без LIKE

10. Выборка, содержащая дубликаты трудна для чтения. Маловероятно, что такой запрос составлен намеренно. Кроме того, зачастую дубликаты сами являются индикаторами существования прочих ошибок, например, отсутствие условий объединения.

Еггг 25: Большое количество дубликатов

11. В программах на С могут быть случаи, когда не присвоен NIL указатель и программа дает сбой. Такие ошибки возможны и в SQL, и вероятность их возникновения должна учитываться. Поскольку эти проблемы зависят от состояния базы данных, их достаточно трудно обнаружить во время теста.

Еггг 28: Подзапрос должен выдать не более одной выборки

Еггг 29: Переменной не может быть присвоено значение 'null'

Еггг 30: Трансформация сложных типов данных

Очевидно, что основные задачи поиска семантических ошибок SQL-запросов достаточно актуальны. В перспективе хотелось бы, чтобы каждая СУБД имела в своем арсенале средства обработки таких запросов, но пока ни одна СУБД не выводит таких предупреждений. Список типов ошибок, содержащихся в данной статье, может служить основой для разработки алгоритмов их выявления.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Stefan Brass, Christian Goldberg. Detecting Logical Errors in SQL Queries. Technical Report, University of Halle, 2004.

2. U. S. Chakravarthy, J. Grant, and J. Minker. Logic-based approach to semantic query optimization. ACM Transactions on Database Systems, 15:162–207, 1990.

3. Wesley W. Chu, Hua Yang, Kuorong Chiang, Michael Minock, Gladys Chow and Chris Larson. Cobase: A scalable and extensible cooperative information system. Journal of Intelligent Information Systems, 1996.

4. Chun-Nan Hsu and Craig A. Knoblock. Using inductive learning to generate rules for semantic query optimization. In Advances in Knowledge Discovery and Data Mining, pages 425–445. AAAI/MIT Press, 1996.

5. Welty C. Correcting user errors in SQL. International Journal of Man-Machine Studies 22:4, 463-477, 1985.

Р.М. РЮМИН

Рязанский государственный радиотехнический университет

ОПТИМИЗАЦИЯ ТЕСТИРОВАНИЯ РАБОТЫ КОМПЬЮТЕРНОЙ СЕТИ

Рассматривается проблема составления минимального плана тестирования передачи данных в сети со статической маршрутизацией.

Статическая маршрутизация используется, как правило, в малых сетях, имеющих относительно стабильную структуру, в частности, в корпоративных. Такой подход имеет свои достоинства и недостатки. Основными достоинствами статической маршрутизации являются:

- лёгкость отладки и конфигурирования в малых сетях;
- отсутствие дополнительных накладных расходов (из-за отсутствия протоколов маршрутизации);
- мгновенная готовность (не требуется интервал для конфигурирования/подстройки);
- низкая нагрузка на процессор маршрутизатора;
- предсказуемость в каждый момент времени.

Один из недостатков сети со статической маршрутизацией в том, что она подвержена влиянию случайных ошибок администратора, поэтому после конфигурирования сети требуется провести тестирование передачи данных между различными подсетями. Очевидно, что проверка передачи данных из каждой подсети в каждую не является оптимальной. Маршруты между некоторыми подсетями могут являться частями других маршрутов и не требовать отдельной проверки. Например, если проверена передача данных из сети А в сеть С на рис. 1., проверять передачу данных из сети В в сеть С излишне, т.к. второй путь полностью входит в первый, который уже был проверен.

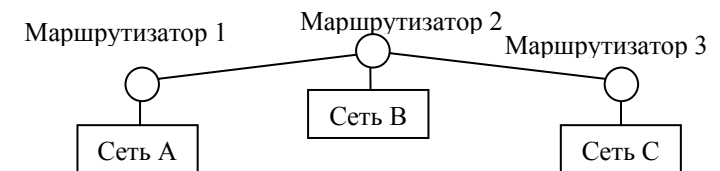


Рис. 1. Пример сети, для которой можно сократить количество проверок при тестировании работы сети

Таким образом, задача заключается в составлении плана тестирования, обеспечивающего полную проверку работоспособности всей сети и при этом включающего минимальное количество операций проверки. Исходными данными задачи являются граф структуры сети $G = (X, \Gamma)$ и матрица маршрутов из каждой подсети в каждую $M = [\mu_{ij}]$. Результатом должно являться множество выполняемых проверок $P = \{(i, j) | 1 \leq i \leq |X|, 1 \leq j \leq |X|, i \neq j\}$, содержащее пары, где первая компонента – номер сети источника данных, а вторая компонента – номер сети приемника данных на одном из этапов проверки. Множество P должно обеспечивать полную проверку всей сети и при этом иметь минимальную мощность.

Для решения этой задачи просматривается матрица M и при каждом просмотре выбирается маршрут, покрывающий наибольшее количество элементарных маршрутов (маршрутов между соседними узлами, длиной в одно ребро), непокрытых ранее. Номера сетей источника и приемника данных выбранного маршрута (номера строки и столбца матрицы) становятся очередной парой множества проверок P . Покрытые этим маршрутом элементарные маршруты вычеркиваются из всех элементов матрицы M . Просмотр матрицы осуществляется до тех пор, пока не будут вычеркнуты все элементарные маршруты каждого элемента.

В результате выполнения указанной эвристической процедуры получаем множество P , которое содержит малое число проверок, позволяющих протестировать работу всей сети.

Н.Е. СКОРОБОГАТОВА, Е.М. ДОНДИК

Рязанский государственный радиотехнический университет

СИСТЕМА РАСПОЗНОВАНИЯ ДАКТИЛЬНОЙ РЕЧИ

Рассматривается система распознавания и звукового воспроизведения дактильной речи, использующая контурное представление знаков.

Распространение информационных технологий в окружающей повседневной жизни ставит вопрос оказания помощи людям с дефектами слуха и речи. Они испытывают значительные трудности в общении с окружающими, в том числе при посещении поликлиник, учреждений социальной сферы и т.д., а, кроме того существуют и проблемы

обучения таких детей дактильной и жестовой речи в детских садах и школах.

Пока из-за сложностей в настоящее время остается не решенной проблема автоматического распознавания жестовой речи. Поэтому задача распознавания и воспроизведения на первых этапах хотя бы дактильной речи может оказать помощь таким людям в общении с окружающим миром.

В основу русской дактильной речи положена одноручная, в отличие от английской двуручной, система представления знаков алфавита и поэтому для распознавания можно выбрать контура бинаризованного изображения. При этом описания контуров используемые в компьютерной графике, таких как кривые Безье, линейно-контурные модели и др. не всегда подходят для реализации алгоритма распознавания. Поэтому в рассматриваемой системе используется кодирование контура элементарными векторами, которые формируют вектор-контур каждой дактилемы [1].

При распознавании знаков дактильной речи (дактилем) возникают ряд специфических проблем, которые должны быть учтены распознающей системой.

Во-первых, знаки демонстрируются в непрерывном движении, без фиксации момента представления буквы, что требует от систем отслеживания направления движения элементов руки и определения факта изменения этого направления. При покадровом анализе факт изменения направления может служить основанием для считывания (или селекции) предыдущего кадра в качестве знакового.

Во-вторых, ряд дактильных знаков демонстрируется не только с фиксацией положения пальцев руки в заданной позиции, но и с колебательными движениями одного пальца. Это существенно затрудняет распознавание таких знаков, по сравнению с фиксированными и приводит к необходимости отличать такие колебания от движения перехода к следующему знаку в дактильной речи.

В-третьих, реальная дактильная речь может демонстрироваться на различном расстоянии от веб-камеры и людьми с различными размерами руки (взрослыми или детьми) поэтому распознающая система должна быть инвариантна к масштабу изображения дактилемы.

В-четвертых, положение руки при демонстрации знака не фиксировано на поле кадра и может смещаться или наклоняться в зависимости от привычки демонстрирующего дактильную речь. Следовательно, распознающая система должна обеспечивать инвариантность по отношению к смещению и повороту распознаваемого знака.

Все приведенные проблемы накладывают на распознающую систему достаточно сложные и отчасти противоречивые требования.

Предлагаемая система распознавания и воспроизведения дактильной речи решает поставленные проблемы и может быть реализована программно на современных компьютерах.

Система воспринимает дактильную речь с помощью видеодатчика (ВК) в цифровом виде (веб-камера или цифровая видеокамера). В связи с тем, что движение руки не фиксируется на отдельном дактильном знаке, а находится в непрерывном переходе от одного знака к другому, то первая задача системы распознавания состоит в определении момента демонстрации конкретного знака. Для этого в системе, приведенной на рис. 1, используется программа селекции знака (ПСЗ), которая из потока изображений движущейся руки и пальцев определяет моменты изменения направления их движения и фиксирует предшествующий кадр в качестве демонстрируемой буквы.

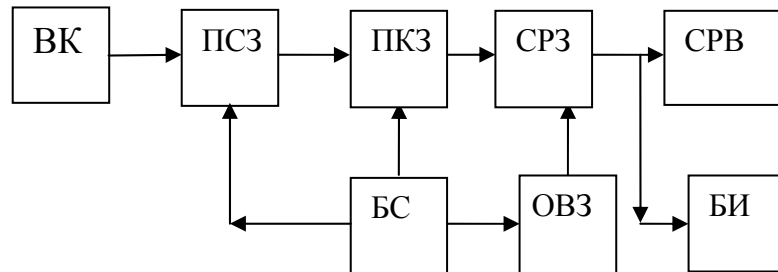


Рис. 1. Структура системы распознавания

Выделенный в растровом формате знак поступает на вход программы кодирования знака (ПКЗ), которая реализует выделение контура изображения знака (буквы) в бинарном представлении, как показано на рис. 2, и осуществляет кодирование кадра. Таким образом, каждый знак представляется в закодированном виде цепочкой векторов в комплексном пространстве, позволяющем учесть направления каждого составляющего вектора. В таком же закодированном виде хранится обучающая выборка знаков (ОВЗ), которая может в определенной мере подстраиваться под различные виды освещения и фона распознаваемых знаков. Система распознавания знаков (СРЗ) осуществляет сравнение демонстрируемого знака с эталонными, предварительно компенсируя различие в масштабе, повороте или смещении и выдает результат на блок индикации (БИ) а также систему речевого воспроизведения (СРВ).

Управление всей системой осуществляет блок синхронизации (БС), который обеспечивает последовательное выполнение операции селекции знака, его кодирование и распознавание.



Рис. 2. Дактилема буквы «У» и ее контур

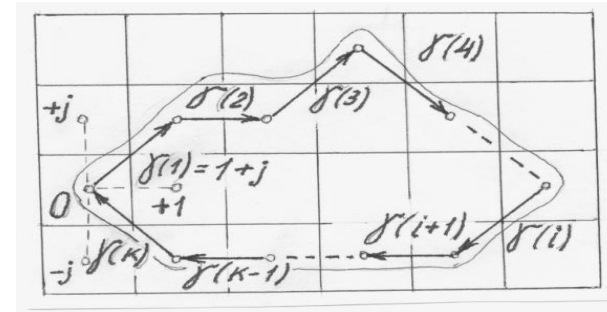


Рис. 3. Представление контура дактилемы элементарными векторами

Общий алгоритм функционирования системы распознавания дактильных знаков включает ряд последовательных процедур, таких как селекция знаков, выделения и кодирование контуров, формирование базы вектор-контуров обучающей выборки, формирование вектор-контура исследуемого знака, реализация процедуры распознавания знака, отображение результата распознавания и звуковое воспроизведение дактилемы.

Ввод знака осуществляется веб-камерой обычно в пок кадровом растровом полутоновом формате и обычно в цветном варианте. Для ускорения обработки выбирается один наиболее информативный монохромный канал изображения и оно переводится путем выбора уровня отсечения яркости пикселей в бинарный формат. В этом формате выделяется контур знака пока еще в растровом представлении. Далее изображение знака приводится в фиксированную координатную точку и путем комплекса преобразований выделяются моменты изменения направлений движения элементов знака. Кадр предшествующий этому

моменту берется в качестве значащего и направляется на систему распознавания.

Каждый контур анализируемого знака Γ кодируется набором элементарных векторов $\gamma(i)$ $i=1, \dots, k$, где k – составляет число элементарных векторов вектор-контра $\Gamma = \{\gamma(i)\}_k$. При этом контур отслеживается окном размером 3×3 пиксела на границе бинарного изображения и фона при заданных логических условиях.

Таким образом, комплекснозначный код контра $\Gamma = \{\gamma(i)\}_k$ будет представлен на растровой плоскости изображения последовательностью векторов как функции от дискретного комплексного переменного, что приведена на рис. 3. Первый элементарный вектор $\gamma(1)$ выходит из начальной точки 0 и направлен в ближайший из 5 (а в общем случае из 8) пикселов с конкретной комплексной координатой $(1 + j)$. Следующий элементарный вектор $\gamma(2)$ принимает за начало отсчета конец вектора $\gamma(1)$ и имеет комплексное направление $(1 + 0j)$. Представленный на рисунке 3 замкнутый вектор-контур Γ в общем случае имеет вид

$$\Gamma = \{ \gamma(1), \dots, \gamma(i), \gamma(i+1) \dots, \gamma(k) \}, \quad (1)$$

так что последний элементарный вектор $\gamma(k)$ заканчивается в точке начала отсчета 0. Принимается по умолчанию, что кодирование контра осуществляется по часовой стрелке.

Так как дискретность комплексного переменного совпадает с шагом растрового представления изображений это существенно упрощает описание и переводит анализ контуров в формат векторного анализа. Для решения задач распознавания в системе должно соблюдаться условие замкнутости контуров. Это означает, что исследуемой контур Γ и эталонный контур V должны соблюдать условие равенства нулю суммы всех составляющих их элементарных векторов

$$\sum_{i=1}^k \gamma(i) = 0 \quad \text{и} \quad \sum_{i=1}^k v(i) = 0. \quad (2)$$

Такое представление контуров последовательностью комплексных чисел, задающих элементарные вектора, предоставляет возможность аналитический операций с контурами. Полное равенство исследуемого Γ и эталонного V векторов обеспечивается при их комплексном представлении

$$\Gamma = \{ \gamma_1(i) + j \gamma_2(i) \} \quad \text{и} \quad V = \{ v_1(i) + j v_2(i) \}, \quad (3)$$

в виде равенств

$$\gamma_1(i) = v_1(i); \quad \gamma_2(i) = v_2(i), \quad i=1, \dots, k. \quad (4)$$

Однако в процессе распознавания обеспечить полную идентичность всей последовательности элементарных векторов затруднительно. Поэтому система распознавания знаков предусматривает возможность смещения начальной точки отсчета (начального пикселя) на d элементов контра. Новые смещенные элементарные вектора $\gamma^{(d)}$ получаются по соотношению

$$\gamma^{(d)} = \gamma(i+d),$$

т.е. номера векторов сдвигаются на d отсчетов.

При этом текущая векторная сумма $\beta(i)$ сумма элементов кода $\gamma(i)$, на i -ом элементе кода контра определяемая как

$$\beta(i) = \sum_{s=0}^i \gamma(s), \quad (5)$$

и формирует другой суммарный код контра $B = \{\beta(i)\}_k$, при $i=1, \dots, k$.

Для замкнутого контра текущие значения суммарного кода должны быть $\beta(1) = \gamma(1)$ и $\beta(k) = 0$.

Код контра Γ и текущий суммарный код B инвариантны к переносу знака дактилемы по полю изображения, что кардинально упрощает решение задачи распознавания.

Кроме того использование в качестве информативных признаков замкнутых вектор-контуров позволяет выполнить распознавание при изменении масштаба знака дактилемы и его наклона. В качестве решающей процедуры распознавания в системе используются нормированные скалярные произведения вектор-контра анализируемой дактилемы Γ и вектор-контра V каждого из числа знаков хранящихся в базе данных.

Скалярное произведение двух векторов Γ и V , представляемых в виде (3), будет определяться соотношением

$$(\Gamma, V) = \sum_{i=1}^k [\gamma_1(i) v_1(i) + \gamma_2(i) v_2(i)] + j \sum_{i=1}^k [\gamma_2(i) v_1(i) - \gamma_1(i) v_2(i)] \quad (6)$$

и является для выбранного способа кодирования контуров комплексным числом.

Нормированное скалярное произведение также будет комплексной величиной

$$p = \frac{(\Gamma, V)}{\|\Gamma\| \|V\|} \quad (7)$$

где $\|\Gamma\|$ и $\|V\|$ – нормы соответствующих вектор-контуров.

Модуль нормированного скалярного произведения и является мерой близости вектор-контуров распознаваемой и эталонной дактилемы. При необходимости сравнения вектор-контуров дактилем различающейся размерности возможно использование их спектрального разложения или вычисления взаимнокорреляционной функции с введением дополнительных нулевых элементарных векторов для обеспечения корректного распознавания.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Фурман Я.А. Согласованная фильтрация контуров изображений // Радиотехника. -1995. -№6. -С.30-33

А.И. ТАГАНОВ

Рязанский государственный радиотехнический университет

МЕТОДЫ ПРЕДСТАВЛЕНИЯ РИСКОВ ПРОЕКТА НА ОСНОВЕ СТРУКТУРНО-СИМВОЛЬНЫХ МОДЕЛЕЙ

Рассматриваются методы структурно-символьного построения библиотеки базовых графических моделей рисков проекта.

Введение

Одним из удобных и эффективных способов коллективного обсуждения рискованных ситуаций проекта является наглядное графическое представление рискованной информации по проекту на компьютерном экране в виде графических контурных моделей, отображающих рискованные поля проекта в виде графических фигур в заданной системе координат [1,2].

Координатами отображения рискованных полей на плоскости (x,y) могут выступать соответственно такие характеристики рисков как: x - этап проекта (время) и y- субъективная вероятность риска; x - субъективная вероятность риска, y - субъективная величина ущерба от риска; x - этап проекта, y - субъективная величина ущерба от риска по характеристикам качества проектируемой программной системы и др. [1,3,5].

Возможности программного способа построения и отображения графических моделей рисков проекта по критериям наглядности и интерактивности во многом определяются количественным и качественным составом базовых графических моделей рисков проекта в электронной библиотеке на основе использования структурно-символьных моделей представления рисков [4]. В связи с этим в статье предлага-

ются новые методы построения библиотеки базовых графических моделей рисков проекта, основанные на использовании структурно-символьных моделей.

1. Прямой структурно-символьный метод представления графических моделей рисков проекта

Графическая модель рисков проекта в виде контурных областей в заданной системе координат на основе структурно-символьного метода может быть задана в следующем виде [4]:

$$M = \{t_j, (x, y)_j, (\xi_K)_j, Q\}, (j = \overline{1, N}). \quad (1)$$

Здесь t_j – код базовой графической модели рискованной ситуации (области) проекта, представленной в виде j-го базового графического элемента (БГЭ) в библиотеке; $(x, y)_j$ - абсолютные координаты центра относительной системы координат j -го БГЭ; $(\xi_K)_j$, $(K = 1, 2, \dots, 1_j)$ - кортеж системных параметров j-го БГЭ (масштаб, код ориентации и др.); $Q=(E,U)$ - граф отношений БГЭ, в котором множеству вершин $E = \{e_1, e_2, \dots, e_N\}$ соответствует множество БГЭ в модели (1), а множеству дуг $U = \{u_1, u_2, \dots, u_L\}$ - множество пар БГЭ, находящихся в определенном теоретико-множественном отношении.

При задании графа Q в виде матрицы весовых соотношений $C = \|C_{ij}\|_{N \times N}$, элемент $C_{ij} = \lambda_{ij}$, если вершины e_j и e_i являются смежными, и $C_{ij} = 0$ в противном случае. Значение λ_{ij} определяет код заданного теоретико-множественного отношения (совместного преобразования) БГЭ между собой из числа операций, поддерживаемых системой обработки и отображения графической информации (например, операции объединения, пересечения, отсечения и дополнения контурных областей) [3].

В частном случае граф Q может иметь структуру дерева выполнения графических операций $Q = \{Q_1, Q_2, \dots, Q_r\}$, где $Q_i = (E_i, \lambda_i)$ содержит кортеж $E_i = (e_{i1}, e_{i2}, \dots, e_{iK_i})$, упорядоченный в соответствии с последовательностью выполнения теоретико-множественных операций $\lambda_i = (\lambda_{i1}, \lambda_{i2}, \dots, \lambda_{iK_i})$.

Из структуры модели (1) можно выделить ряд ее особенностей:

- позволяет задавать графические модели рисков проекта достаточно высокой сложности посредством программного синтеза контурных моделей рискованных областей проекта на основе открытой библиотеки БГЭ;

- модель имеет достаточно простую и компактную структуру и не требует значительных объемов памяти для своего хранения;

- наличие в модели графа Q и кортежей системных параметров значительно повышает адаптируемость и гибкость модели в процессе настройки системы программного синтеза на отображение контурных моделей рисков областей требуемой информативности и графической сложности;

- математическая модель БГЭ в библиотеке однозначно может быть задана в виде:

$$M_j = (K_j, G_j, E_j, L_j), \quad (2)$$

где K_j - множество контуров j -го БГЭ; G_j - множество описаний, задающее взаимное расположение элементов множества K_j ; L_j - множество описаний градаций, используемых при построении j -го БГЭ; E_j - множество описаний, задающих отображение $E_j: K_j \rightarrow L_j$.

2. Лингвистический метод представления алфавита базовых графических моделей рисков проекта

Представим математическую модель библиотеки базовых графических моделей рисков проекта в виде алфавита, в котором каждый символ представляет собой базовый графический элемент:

$$M_\delta = \{M_j\}, (j = \overline{1, N}), \quad (3)$$

где N - длина алфавита, под которой понимается количество БГЭ в библиотеке.

Компонуя элементы множеств $M_j, (j = \overline{1, N})$, по функциональному признаку, запишем выражение (3) в виде:

$$M_\delta = (K_\delta, G_\delta, E_\delta, L_\delta),$$

где $K_\delta = \{K_j\}$; $G_\delta = \{G_j\}$; $L_\delta = \{L_j\}$;

$$(\forall i, j \in N, i \neq j)(K_i \cap K_j = \emptyset).$$

Рассматривая всю совокупность множества контуров, входящих в K_δ , всегда можно выделить некоторое подмножество контуров $K(A)$, на основе которого строятся подмножества $K_j(A)$, необходимые для описания каждого j -го БГЭ из заданного алфавита библиотеки:

$$K(A) = \{k_1(A), k_2(A), \dots, k_i(A), \dots, k_n(A)\} = \bigcup_{i=1}^n k_i(A),$$

где $k_1(A), k_2(A), \dots, k_i(A), \dots, k_n(A)$ - есть контуры, абсолютные описания которых привязаны к системе координат БГЭ в библиотеке.

В соответствии с лингвистическим способом совокупность элементов множества K_δ предлагается рассматривать как некоторый формальный язык. Каждое отдельное подмножество $K_j(A) \in K_\delta, (j = \overline{1, N})$ этого языка можно представить "фразой", составленной из ряда "слов", а "словом" считать контур. "Грамматика" такого формального языка должна содержать набор правил, обеспечивающих однозначное представление всех "фраз" языка на основе "слов". В рассматриваемом нами случае грамматика содержит указания, какие контуры входят в состав синтезируемых множеств $K_j \in K_\delta, (j = \overline{1, N})$.

Давая каждому из абсолютных описаний контуров $k_i(A) \in K_j(A), (i = \overline{1, n}; j = \overline{1, N})$ относительные описания, например порядковые номера, и указывая, какие относительные описания содержат элементы множества $K_j(A)$, мы тем самым устанавливаем грамматические правила синтеза элементов множества K_δ .

В этом случае вместо абсолютных описаний элементы множества $K_j(A)$ содержат уже относительные описания, что более компактно и гибко представляет структурное построение библиотеки БГЭ рисков полей в виде:

$$K_j(O) = \{1^{\alpha_{j1}}, 2^{\alpha_{j2}}, \dots, n^{\alpha_{jn}}\} = \bigcup_{i=1}^n i^{\alpha_{ji}},$$

где

$$i^{\alpha_{ji}} = \begin{cases} i, & \text{при } \alpha_{ji} = 1 \\ 0, & \text{при } \alpha_{ji} = 0 \end{cases},$$

i - относительные описания контуров $k_i(O) \in K(O), (i = \overline{1, n})$.

Такое преобразование описаний позволяет организовать структуру контурной информации в библиотеке БГЭ на основе двух множеств:

$$K_\delta(A) = \{K_\delta(O), K(A)\} \quad (4)$$

где при использовании обозначений в форме условного суммирования подмножеств контуров и объединения относительных описаний контуров имеем:

$$K_{\delta}(O) = \{K_j(O)\} = \sum_{j=1}^N K_j(O) = \sum_{j=1}^N \bigcup_{i=1}^n V_i^{\alpha_{ji}}. \quad (5)$$

Из (4) и (5) следует, что исключается многократное дублирование абсолютных описаний контуров в описании множества $K(A)$, и хранятся в библиотеке два множества описаний, включающие в себя соответственно абсолютные описания всех оригинальных контуров и относительные описания $K_{\delta}(O)$, задающие состав каждого БГЭ в библиотеке.

Информационный объем описаний множества K_{δ} по грамматическому принципу определится как:

$$V^{(2)}(A) = \sum_{j=1}^N \sum_{i=1}^n V_i^{\alpha_{ji}}(O) + \sum_{i=1}^n V_i(A), \quad (6)$$

где второе слагаемое в (6) отражает информационный объем абсолютных описаний всех контуров множества, а первое слагаемое является информационным объемом уже относительных описаний элементов множества $K_{\delta}(O)$ в модели (4):

$$V(O) = \sum_{j=1}^N \sum_{i=1}^n V_i^{\alpha_{ji}}(O).$$

Из выражения (6) видно, что по критерию требуемого информационного объема эффективность лингвистического способа задания множества K_{δ} возрастает в тех случаях, когда элементы множества $K(A)$ многократно используются при задании элементов $K_j \in K_{\delta}$ и когда абсолютные описания множества $K(A)$ требуют более значительного объема памяти для хранения в библиотеке по сравнению с их относительными описаниями.

3. Программный метод представления алфавита базовых графических моделей рисков проекта

Рассмотрим следующий предлагаемый способ задания множества $K(A)$ в модели (4), основанный на программном синтезе его элементов. Будем считать, что в целях сокращения числа хранимых абсолютных описаний контуров в библиотеке БГЭ удалось выделить некоторое множество $K'(A) \in K(A)$ такое, что $K_p'(A) = K(A) \setminus K'(A)$ синтезируются на основе $K'(A)$ программными средствами графической системы, т.е.

$$K(A) \equiv K_p(A) = \{K_p'(A), K'(A)\},$$

где $K_p(A)$ - множество требуемых абсолютных описаний контуров; $K'(A)$ - множество хранящихся абсолютных описаний контуров; $K_p'(A)$ - множество описаний программно-синтезируемых на основе $K'(A)$.

Для упрощения дальнейших рассуждений условно можно записать:

$$K_p(A) = \{K_{p_1}(A)\} = \sum_{l=1}^n K_{p_1}(A),$$

где $K_{p_1}(A)$ - информационное описание, достаточное для программного синтеза контура $k_1(A)$; n - количество контуров, необходимое для построения всего алфавита библиотеки.

По аналогии с (6) введем обозначения:

$$k_t^{\beta}(A) = \begin{cases} k_t(A), & \text{при } \beta = 1; \\ 0, & \text{при } \beta = 0. \end{cases}$$

Тогда для синтеза абсолютного описания l -го контура следует выполнить:

$$K_{p_1}(A) = \{k_1^{\beta_{11}}(A), k_2^{\beta_{12}}(A), \dots, k_n^{\beta_{1n}}(A), p_1\} = \\ = \left\{ \bigcup_{t=1}^{n'} k_t^{\beta_{1t}}(A), p_1 \right\},$$

где β_{lf} принимает значения 1 и 0 в зависимости от участия того или иного контура при программном синтезе контура k_l ; p_1 - множество кодов операций программного синтеза с использованием элементов множества

$$k_l'(A) = \bigcup_{t=1}^{n'} k_t^{\beta_{lt}}(A).$$

Определим информационный объем описаний, необходимый для программного синтеза всей совокупности контуров K_{δ} . Для этого введем обозначения [3]: $V_t(A)$ - информационный объем абсолютного описания контура $k_t'(A)$; $V_{p_1}(A)$ - информационный объем абсолютного описания кодов операций для программного синтеза $k_t(A)$.

Тогда информационный объем множества $K_{p_1}(A)$ будет равен:

$$V_{p_1}(A) = \sum_{t=1}^{n'} V_t^{\beta_{1t}}(A) + V_{p_1}(A),$$

и соответственно информационный объем множества $K_p(A)$ определится как:

$$V_{K_p}^{(3)}(A) = \sum_{l=1}^n \left(\sum_{t=1}^{n'} V_t^{\beta_{lt}}(A) + V_{p_l}(A) \right), \quad (7)$$

Выражение (7) на основании данных таблицы 3 преобразуем к виду:

$$V_{K_p}^{(3)}(A) = \sum_{t=1}^{n'} V_t(A) r_t + \sum_{l=1}^n V_{p_l}(A). \quad (8)$$

Из выражения (8) видно, что информационный объем $V_{K_p}(A)$ определяется числом контуров n' , информационным объемом абсолютного описания каждого контура $k_t'(A) \in K'(A)$, частотой r_t использования описаний каждого контура при синтезе других контуров, а также информационным объемом хранения кодов операций V_{p_l} .

4. Программно-лингвистический метод представления алфавита базовых графических моделей рисков проекта

Здесь вместо абсолютных информационных описаний $K_{p_l}(A)$ введем также относительные описания $K_{p_l}(O)$, указывающие, где хранятся абсолютные описания. Такое разделение позволит хранить без повторов всю совокупность абсолютных описаний $K(A)$ и отдельно хранить с возможными повторениями уже относительные описания в виде:

$$K_p(A) = \{K_p'(O), K'(A)\}, \quad (9)$$

где $K_p'(O)$ - множество задано в форме:

$$K_p'(O) = \{K_{p_l}'(O)\} = \sum_{l=1}^n K_{p_l}'(O),$$

а множество относительных описаний определится как:

$$K(O) = \{ \{1^{\beta_{11}}, 2^{\beta_{12}}, \dots, t^{\beta_{1t}}, \dots, n^{\beta_{1n'}}\} \} = \{ \sum_{t=1}^{n'} t^{\beta_{lt}}(A), p_l \}.$$

Получим выражение для информационного объема множества $K_p(A)$ при программно-лингвистическом методе его представления.

На основании (9) имеем:

$$V_{K_p}^{(4)}(A) = \sum_{l=1}^n \left(\sum_{t=1}^{n'} V_t^{\beta_{lt}}(O) + V_{p_l}(A) \right) + \sum_{t=1}^{n'} V_t(A),$$

или в другом виде через значения $R = \{r_1, r_2, \dots, r_{n'}\}$,

$$V_{K_p}^{(4)}(A) = \sum_{t=1}^{n'} (V_t(O) r_t + V_t(A)) + \sum_{l=1}^n V_{p_l}(A).$$

Для оценки соотношения величин $V_{K_p}^{(3)}(A)$ и $V_{K_p}^{(4)}(A)$ рассмотрим их отношение:

$$\eta_{(4,3)} = \frac{V_{K_p}^{(4)}}{V_{K_p}^{(3)}} = \frac{\sum_{t=1}^{n'} (V_t(O) r_t + V_t(A)) + \sum_{l=1}^n V_{p_l}(A)}{\sum_{t=1}^{n'} V_t(A) r_t + \sum_{l=1}^n V_{p_l}(A)}. \quad (10)$$

Из выражения (10) следует, что эффективность программно-лингвистического метода, с учетом структурно-символьного представления базовых графических моделей рисков проекта в библиотеке, по сравнению с программным методом выше лишь в случае многократного использования одних и тех же контурных описаний при программном синтезе на их основе алфавита БГЭ в библиотеке. При этом провести аналитическое сравнение первого прямого метода и второго лингвистического с третьим программным и четвертым программно-лингвистическим методами весьма сложно, что объясняется их качественным различием. Однако является очевидным, что программно-ориентированные методы имеют большие возможности по гибкости и компактности библиотечного представления широкого перечня базовых графических моделей рисков проекта, необходимых для функционирования интерактивной графической системы отображения рисков полей проекта с высоким уровнем сервисов на этапе процесса идентификации и анализа рисков проекта экспертными методами.

Заключение

Таким образом, предлагаемые в работе методы построения базовых графических моделей рисков проекта расширяют функциональные возможности графических интерактивных систем для отображения рисков полей проекта по многим параметрам рисков [1,3,5] посредством гибкой настройки алфавита библиотеки базовых графических моделей рисков на заданный класс идентифицируемых и анализируемых рисков. При этом для гибкой настройки и оптимизации структуры построения алфавита библиотеки по ресурсным и временным критериям могут быть сформулированы соответствующие оптимизационные задачи [3,4], для решения которых в условиях лингвистической неопределенности могут быть использованы и исследованы подходы, основанные на методах теории нечетких множеств [5] и на методах теории неопределенного программирования [6].

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. ANSI/PMI 99-001-2004. Руководство к Своду знаний по управлению проектами (Руководство РМВОК). - 388 с.
2. Таганов А.И. Построение изображений физических полей на основе декомпозиции контурной информации: Межвуз. сборник научных трудов «Методы и приборы контроля параметров биосферы. - Л.: ЛИАП, 1984. Вып. 171. С. 90-94.
3. Липаев В.В. Анализ и сокращение рисков проектов сложных программных средств. – М.: СИНТЕГ, 2005. – 224 с.
4. Таганов А.И. Методы представления сложной структурно-символьной информации // Научно-технический журнал «Вестник РГРТА», 2006. Вып. 18. С. 74 -80.
5. Таганов А.И. Применение нечетких множеств для формализации процессов анализа и идентификации важности рисков программного проекта // Научно-технический журнал «Системы управления и информационные технологии», 2007. № 4(30). С. 46-51.
6. Теория и практика неопределенного программирования / Б.Лю; пер. с англ. – М.: БИНОМ. Лаборатория знаний, 2009. – 416 с.

А.И. ТАГАНОВ

Рязанский государственный радиотехнический университет

МЕТОД СИНТЕЗА ЛОГИКО-АЛГЕБРАИЧЕСКИХ МОДЕЛЕЙ РИСКОВ ПРОЕКТА

Рассматривается специальный метод синтеза логико-алгебраических моделей рисков проекта, основанный структурно-символьном исходном описании рисков и теоретико-множественных операциях синтеза контурных моделей рисков полей проекта.

Введение

Важной задачей процесса управления рисками проекта является системный анализ предметной области проекта с целью идентификации потенциальных угроз проекту. Для решения такой трудноформализуемой задачи используются методы мозгового штурма, в процессе применения которых эксперты по рискам принимают коллективные решения по идентификации рисков, используя для анализа рисков в том числе ту рисковую информацию, которая визуально представлена в различных системах координат на компьютерном экране в виде рискованных полей проекта [1,2].

Для расширения функциональных возможностей графических методов представления, моделирования и отображения сложной рисковой информации о проекте в настоящей работе предлагается оригинальный математический аппарат синтеза логико-алгебраических моделей рисков проекта, основанный специальных теоретико-множественных операциях синтеза контурных графических моделей рискованных областей проекта с учетом структурно-символьного способа описания этих полей [3].

Для построения предлагаемого метода синтеза логико-алгебраических моделей рисков проекта представим вначале на формальном уровне необходимые теоретико-множественные операции (ТМО) объединения, пересечения, отсечения и дополнения графических объектов, представляющих собой односвязные области рискованных полей проекта, ограниченные жордановыми кривыми (контурами) произвольной сложности. Затем обобщим эти результаты на случаи выполнения ТМО над многосвязными рискованными областями. При этом ограничимся рассмотрением такой информационной достаточности ТМО, которая необходима для отображения результатов геометрического моделирования рисков проекта растровым способом [1,3].

Теоретический анализ и разработка метода

Логико-алгебраическую модель рискованных полей проекта, описываемую структурно-символьным методом, будем представлять здесь в виде [3]:

$$MM_{вк} = \{t_j, (x, y)_j, (\xi_k)_j, Q\}, (j = \overline{1, N}) \quad (1)$$

Здесь t_j - код j -го базового элемента (БЭ) в библиотеке; $(x, y)_j$ - абсолютные координаты центра привязочной системы координат j -го БЭ; $(\xi_k)_j, (K=1, 2, \dots, l_j)$ - кортеж системных параметров j -го БЭ (масштаб, код ориентации и др.); $Q=(E, U)$ - граф отношений БЭ между собой, в котором множеству вершин $E = \{e_1, e_2, \dots, e_N\}$ поставлено в соответствие множество БЭ в модели (1), а множеству дуг $U = \{u_1, u_2, \dots, u_L\}$ - множество пар контактируемых между собой БЭ. При задании графа Q в виде матрицы весовых соотношений $C = \|C_{ij}\|_{N \times N}$, элемент $C_{ij} = \lambda_{ij}$, если e_j смежна e_i и $C_{ij} = 0$, если e_j не смежна e_i . Значение λ_{ij} определяет некоторое контурное теоретико-множественное соотношение $\chi(\lambda_{ij})$ из заданного набора допустимых отношений БЭ между собой.

Основная решаемая здесь проблема заключается в разработке необходимых теоретических положений предлагаемого метода, позво-

ляющих теоретически обоснованное построение алгоритмов синтеза логико-алгебраических моделей контурных изображений рисковых полей проекта.

Введем в рассмотрение множество точек $T^2 = (t_1, t_2, \dots, t_N)$ полученных в результате работы оператора $Sch(K_1, t_0, \overrightarrow{K_2})$ [2], аргументы которого K_1, t_0 и $\overrightarrow{K_2}$ представляют собой соответственно контур K_1 области $P_1, \overrightarrow{K_2}$ - ориентированный контур области P_2 и точка t_0 , удовлетворяющая условиям $t_0 \in K_2 \& t_0 \notin P_1$.

Выделенное таким оператором множество точек T^2 обладает следующими очевидными и важными свойствами:

Свойство 1. Множество T^2 содержит всегда четное число точек.

Свойство 2. Множество точек T^2 делит контур K^2 на множество отрезков $t_0 \in K_2 \& t_0 \notin P_1$, где $t_0 \in K_2 \& t_0 \notin P_1, h_n^2 = (t_n, t_1)$;

$\bigcap_{j=1}^n h_j^2 = \emptyset; \bigcup_{j=1}^n h_j^2 = K^2$, которые линейно упорядочены относительно t_0 в

направлении обхода контура $\overrightarrow{K_2}$, т.е. $\forall i \in (1, n)(h_1^2 < h_2^2 < \dots, h_n^2)$.

Свойство 3. Множество H^2 всегда можно разделить на два упорядоченных подмножества A^2 и B^2 ($A^2 = (a_1^2, a_2^2, \dots, a_j^2, \dots, a_{n/2}^2)$, где $a_j^2 = h_i^2, i = 2j-1$; $B^2 = (B_1^2, B_2^2, \dots, B_j^2, \dots, B_{n/2}^2)$,

где $a_j^2 = h_i^2 \in H^2, i = 2j$; $A^2 \cup B^2 = H^2; A^2 \cap B^2 = \emptyset$);

таких, что $\forall ij \in (1, n/2)[(a_i^2 \in P_1) \& (B_j^2 \setminus t_j = \inf(B_j^2) \notin P_1)]$.

Свойство 4. На основании того, что $\forall t_i \in T[\exists d_\xi^2 \in A^2 \& \exists B_\eta^2 \in B^2 (\inf(a_\xi^2) = \sup(B_\eta^2) = t_i \vee \sup(a_\xi^2) = \inf(B_\eta^2) = t_i)]$,

может быть построен отрезок

$c^2 = a_\xi^2 \cup B_\eta^2 : [(\inf(c^2) = \inf(a_\xi^2) \& \sup(B_\eta^2) \vee (\inf(c^2) = \inf(B_\eta^2) \& \sup(a_\xi^2) = \sup(B_\eta^2))]$.

Следствие 1. На основании свойства 4 может быть построена процедура, выделяющая из множества отрезков Q такие подмножества отрезков $Q_i \in Q$, что каждое подмножество Q_i образует замкнутый контур K_i .

Для упрощения формализации алгоритмов выполнения операций объединения, пересечения и отсечения плоских объектов, введем в рассмотрение вспомогательный оператор $Ord(T, t_H, \vec{f}) \rightarrow T'$, упорядо-

чивающий элементы множества T в направлении \vec{f} относительно точки $t_H \in T \& t_H \in f$. Здесь $\forall t'_i \in T'[t_i \in f], |T'| = n, \forall i \in (1, n-1) [t'_i < t'_{i+1}]$.

Аналогично множеству T^2 может быть получено другое множество $Sch(K^2, t_0, \overrightarrow{K_1}) \rightarrow T^1$, которое отличается от T^2 лишь упорядоченностью точек $t'_i \in T^1$ относительно t_0 в направлении $\overrightarrow{K_2}$. При этом для T^1 будут справедливы все свойства, отмеченные для множества T^2 .

Тогда алгоритмы выполнения теоретико-множественных операций синтеза рисковых полей проекта на основе рисковых полей, заданных областями P^1 и P^2 ограниченными контурами K^1 и K^2 , будут основываться на следующих теоремах.

Теорема 1. Для выполнения операции объединения двух односвязных рисковых областей проекта P^1 и P^2 , ограниченных жордановыми контурами K^1 и K^2 , информационно достаточным по растровому способу построения является объединение лишь множества отрезков $V^1 \in K^1$ и $V^2 \in K^2$.

Доказательство. Согласно свойства 4 множество $Q = V^1 \in V^2$ всегда может быть однозначно разбито на подмножества $Q_i \in Q = \{Q_1, Q_2, \dots, Q_r\}$ такие, что каждое Q_i образует замкнутый контур K_i^3 не имеющий общих точек с контурами $K_j^3, \{j \neq i, j = \overline{1, r}\}$. В этом случае доказано, что множество $K^3 = \{K_1^3, K_2^3, \dots, K_r^3\}$ является информационно достаточным по растровому способу построения односвязных и многосвязных областей [2].

Кроме того, множество K^3 содержит в себе множества V^1 и V^2 такие, что $V^1 \in P_1 \& V^1 \notin P^2$ и $V^2 \in P^2 \& V^2 \notin P^1$, а это доказывает то, что множество отрезков Q ограничивает область $P^3 = P^1 \cup P^2$.

Следствие 2. Построение на основе Q множества контуров $K^3 = \{K_1^3, K_2^3, \dots, K_r^3\}$ и затем выделение такого $K_\xi^3 \in K^3$, что $\forall i \in \{1, 2, \dots, r\} \setminus \xi [P_\xi^3 \cap P_i^3 = P_i^3]$ является информационно достаточным для построения на основе $(K_\xi^3, (K_1^3, K_2^3, \dots, K_r^3) \setminus K_\xi^3)$ логико-алгебраической модели рисковых полей проекта (1).

В самом деле, выделение из Q множества K^3 эквивалентно заданию в модели (1) множества K . Далее, выделение из множества K^3 внешнего контура K_ξ^3 эквивалентно заданию в модели (1) дерева G^3 и

вследствие двуградиантности рисковых областей P^1 и P^2 информация вида E и L содержится в G^3 .

Следствие 3. Дерево вложения контуров G^3 всегда содержит один либо два уровня вложения множества контуров K^3 , что непосредственно следует из свойства 4 и следствия 1.

На основании теоремы 1 и вышеуказанных свойств может быть построен следующий алгоритм функционирования оператора теоретико-множественного объединения рисковых полей проекта на основе их контурного задания $K^1 \cup K^2 \rightarrow K^3$ и выделяющего множество контуров информационно достаточное по растровому способу построения рисковых областей:

п°1. Выделение точек $t_0 \in K^2$ & $t_0 \notin P^1$ и $t'_0 \in K^1$ & $t'_0 \notin P^2$.

п°2. $Sch(K^1, t_0, \overline{K^2}) \rightarrow T^2; T^2 \rightarrow H^2 = \{A^2, B^2\}$.

п°3. $Ord(T^2, t_0, K^1) \rightarrow T^1; T^1 \rightarrow H^1 = \{A^1, B^1\}$.

п°4. $Q = B^1 \cup B^2$.

п°5. Построение множества K^3 на основе Q.

п°6. Конец алгоритма.

Заметим, что предложенный алгоритм синтеза инвариантен к форме, сложности, способу кодирования контуров и базируется на простейших процедурах Sch и Ord, которые многократно используются в других блоках системы обработки структурно-символьной информации [4].

Теорема 2. Для выполнения теоретико-множественной операции пересечения двух односвязных рисковых областей P^1 и P^2 , ограниченных жордановыми контурами K^1 и K^2 , информационно достаточным по растровому способу построения является объединение лишь множества отрезков K^1 и K^2 .

Доказательство. По аналогии с доказательством теоремы 1 имеем

$$Q = A^1 \cup A^2, Q = \{Q_1, Q_2, \dots, Q_r\}, \text{ где } \forall i \in (\overline{1, r}) (\bigcup_{j=1}^{n_i} q_{ij} = K_i^3).$$

Тогда вследствие того, что множество K^3 содержит отрезки A^1 и A^2 такие, что $A^1 \in P^1$ & $A^1 \in P^2$ и $A^2 \in P^2$ & $A^2 \in P^1$ следует, что множество $Q = A^1 \cup A^2$ ограничивает рисковую область $P^3 = P^1 \cap P^2$.

Следствие 4. В результате выполнения операции пересечения контуров K^1 и K^2 (пересечения областей P^1 и P^2) результат представляет область P^3 , которая является либо односвязной, либо состоит из

множества $P^3 = \{P_1^3, P_2^3, \dots, P_r^3\}$ односвязных областей одной градации ($P_i^3 \cap P_j^3 = \emptyset, i \neq j; i, j = \overline{1, r}$). При этом дерево вложения контуров G^3 содержит все множество $K^3 = \{K_1^3, K_2^3, \dots, K_r^3\}$ на его первом ярусе.

Примечание 1. Алгоритм оператора $K^1 \cap K^2 \rightarrow K^3$ отличается от алгоритма оператора $K^1 \cup K^2 \rightarrow K^3$ лишь тем, что здесь п°4

$$Q = A^1 \cup A^2.$$

Теорема 3. Для выполнения теоретико-множественной операции отсечения односвязной области P^2 от односвязной области P^1 , ограниченных жордановыми контурами K^1 и K^2 , информационно достаточным по растровому способу построения является объединение лишь множества отрезков B^1 и A^2 .

Доказательство. Множество отрезков $Q = B^1 \cup A^2$ по аналогии с теоремами 1 и 2 может быть разбито на подмножества $Q_i \in Q$ такие, что $\forall i \in (\overline{1, r}) [\bigcup_{j=1}^{n_i} q_{ij} = K_i^3] \& \& (K_i^3 \cap K_j^3 = \emptyset, i \neq j, j = \overline{1, r})$.

Тогда на основании того, что множество K^3 содержит множества отрезков B^1 и A^2 такие, что $B^1 \in P^1$ & $B^1 \notin P^2$ и $A^2 \in P^2$ & $A^2 \in P^1$ следует, что множество $Q = B^1 \cup A^2$ ограничивает рисковую область $P^3 = P^1 \setminus P^2$.

Примечание 2. Следствие 1, вытекающее из теоремы 2, справедливо и для теоремы 3.

Примечание 3. Алгоритм оператора $K^1 \setminus K^2 \rightarrow K^3$ имеет отличие от алгоритма, оператора $K^1 \cup K^2 \rightarrow K^3$ лишь в том, что здесь пункт п°4. $Q = B^1 \cup A^2$.

Теорема 4. Для выполнения теоретико-множественной операции отсечения односвязной области P^1 от односвязной области P^2 , ограниченной жордановыми контурами K^1 и K^2 , информационно достаточным по растровому способу построения является объединение лишь множества отрезков B^2 и A^1 .

Доказательство теоремы 4 аналогично доказательству теоремы 3.

Примечание 4. Алгоритм оператора $K^2 \setminus K^1 \rightarrow K^3$ имеет отличие от алгоритма $K^1 \cup K^2 \rightarrow K^3$ лишь в том, что здесь п°4. $Q = A^1 \cup B^2$.

Введем далее понятие контура кадра изображения рисковых полей проекта и понятие дополнения контура изображения рисковых полей.

Определение 1. Если в некотором рассмотрении участвуют только контуры рискованных областей, вложенные в некоторый фиксированный контур I , ограничивающий рискованную область P , то этот самый "большой" контур I называется контуром кадра изображения рискованных полей.

Определение 2. Под дополнением контура K^1 до контура кадра изображения рискованных полей будет понимать некоторую область, расположенную на области кадра изображения и ограниченную множеством контуров $K = \{I, K^1\}$, которое определяется на основе соотношения $\bar{K}^1 = I \setminus K^1 = \{I, K^1\}$. (2)

Следствие 4. Исходя из определения 10, рискованная область, ограниченная множеством контуров \bar{K}^1 , всегда является многосвязной.

Контурные операции теоретико-множественного синтеза над множествами односвязных рискованных областей P^1 и P^2 , где $P_\xi^2 \cap P_\eta^2 = \emptyset$, ($\xi = \eta$); ($\xi, \eta = 1, 2, \dots, m$); $K^1 = \{K_1^1, K_2^1, \dots, K_n^1\}$
 $P^1 = \{P_1^1, P_2^1, \dots, P_n^1\}$, $P_i^1 \cap P_j^1 = \emptyset$, ($i \neq j$); ($i, j = 1, 2, \dots, n$);
 $P^2 = \{P_1^2, P_2^2, \dots, P_m^2\}$, ограничивает $P^1, K^2 = \{K_1^2, K_2^2, \dots, K_m^2\}$ - ограничивает область P^2 , и в дальнейшем будем понимать и записывать в виде:

1. Объединение рискованных областей P^1 и P^2

$$K^1 \cup K^2 = (K_1^1, K_2^1, \dots, K_n^1) \cup (K_1^2, K_2^2, \dots, K_m^2) = \dots((K_1^1 \cup K_1^2) \cup K_2^2) \cup \dots \cup K_m^2 \cup \dots((K_n^1 \cup K_1^2) \cup K_2^2) \cup \dots \cup K_m^2 \cup \dots \cup K_n^1 \cup \dots$$

$$\cup K_m^2 = \bigcup_{i=(1,n)} K_i^1 \cup \bigcup_{j=(1,m)} K_j^2. \quad (3)$$

2. Пересечение рискованных областей P^1 и P^2

$$K^1 \cap K^2 = (K_1^1, K_2^1, \dots, K_n^1) \cap (K_1^2, K_2^2, \dots, K_m^2) =$$

$$= \bigcup_{i=(1,n)} K_i^1 \cap \bigcup_{j=(1,m)} K_j^2. \quad (4)$$

3. Отсечение рискованной области P^2 от рискованной области P^1

$$K^1 \setminus K^2 = (K_1^1, K_2^1, \dots, K_n^1) \setminus (K_1^2, K_2^2, \dots, K_m^2) =$$

$$= \bigcup_{i=(1,n)} K_i^1 \setminus \bigcup_{j=(1,m)} K_j^2. \quad (5)$$

4. Отсечение рискованной области P^1 от рискованной области P^2

$$K^2 \setminus K^1 = (K_1^2, K_2^2, \dots, K_m^2) \setminus (K_1^1, K_2^1, \dots, K_n^1) =$$

$$= \bigcup_{i=(1,m)} K_i^2 \setminus \bigcup_{j=(1,n)} K_j^1. \quad (6)$$

5. Дополнение рискованных областей P^1 и P^2

$$\bar{K}^1 = \{K_1^1, K_2^1, \dots, K_n^1\} = \{I, K_1^1, K_2^1, \dots, K_n^1\}, \quad (7)$$

$$\bar{K}^2 = \{K_1^2, K_2^2, \dots, K_m^2\} = \{I, K_1^2, K_2^2, \dots, K_m^2\}, \quad (8)$$

Таким образом, соотношения (3) ÷ (6) образуют полный набор теоретико-множественных операций синтеза, позволяющий на их основе составлять различные контурные логико-алгебраические модели изображений рискованных полей проекта, которые могут быть преобразованы к требуемому виду на основе следующих (приводимых без доказательства) тождеств алгебры множеств в контурном представлении [4]:

$$1. (K^1 \cup K^2) \cap K^3 = (K^1 \cap K^3) \cup (K^2 \cap K^3);$$

$$2. (K^1 \cap K^2) \cup K^3 = (K^1 \cup K^3) \cap (K^2 \cup K^3);$$

$$3. K^1 \cup K^2 = \overline{\bar{K}^1 \cap \bar{K}^2};$$

$$4. K^1 \cap K^2 = \overline{\bar{K}^1 \cup \bar{K}^2}; \quad (9)$$

Наличие тождеств (9) позволяет на практике осуществлять предварительную обработку входной логико-алгебраической модели рискованных полей проекта с целью ее адаптации к возможностям конкретной системы переработки структурно-символьной информации. Указанное свойство способствует решению проблемы переработки больших объемов контурной информации в режиме реального времени в системах отображения рискованных полей проекта растровым способом.

Рассмотрим далее вопросы построения контурных теоретико-множественных операций объединения, пересечения, отсечения рискованных областей P^1 и P^2 , где P^1 - многосвязная область, а P^2 - односвязная область. Разработка такого математического аппарата позволит на базе односвязных БЭ последовательно формировать внутренние контурные логико-алгебраические модели адекватные некоторым классам изображений рискованных полей проекта, описываемых структурно-символьным методом.

Под многосвязной областью здесь будем понимать область картинной плоскости, ограниченную множеством непересекающихся жордановых контуров $K = \{K_1, K_2, \dots, K_N\}$, один из которых $K_1 \in K$ внешний, а остальные внутренние и расположены на втором ярусе древа вложения контуров.

Назовем окрестностью контура K_i множество $\Gamma^v(K_i)$ элементов дерева вложения контуров, предшествующих контуру K_i и расположенных v -м ярусе ДВК. Обозначим также через $\Gamma^{-1}(K_i)$ узел непосредственно предшествующий узлу K_i .

Утверждение 1. Для выполнения теоретико-множественной операции объединения многосвязной рискованной области P^1 , ограниченной множеством жордановых контуров $K^1 = \{K_1^1, K_2^1, \dots, K_n^1\}$ и односвязной рискованной области P^2 , ограниченной жордановым контуром K^2 , информационно достаточным по растровому способу построения является выполнение контурного соотношения (КС) вида:

$$K^3 = (K_1^1 \cup K_1^2) \cup \left(\bigcup_{K_i^1 \in \Gamma^2(K_1^2)} (K_i^1 \setminus K_i^2) \right). \quad (10)$$

Утверждение 2. Для выполнения теоретико-множественной операции пересечения многосвязной рискованной области P^1 , ограниченной множеством жордановых контуров $K^1 = \{K_1^1, K_2^1, \dots, K_n^1\}$, и односвязной рискованной области P^2 , ограниченной жордановым контуром K^2 , информационно достаточной по растровому способу построения является логико-алгебраическая модель вида:

$$K^3 = (K_1^1 \cap K_1^2) \cup \left(\bigcup_{K_i^1 \in \Gamma^1(K_1^2)} K_i^1 \right). \quad (11)$$

Утверждение 3. Для выполнения теоретико-множественной операции отсечения односвязной рискованной области P^2 , ограниченной жордановым контуром K^2 , от многосвязной рискованной области P^1 , ограниченной множеством жордановых контуров $K^1 = \{K_1^1, K_2^1, \dots, K_n^1\}$, информационно достаточной по растровому способу построения является логико-алгебраическая модель вида:

$$K^3 = (K_1^1 \setminus K_1^2) \cup \left(\bigcup_{K_i^1 \in \Gamma^1(K_1^2)} K_i^1 \right). \quad (12)$$

Утверждение 4. Для выполнения теоретико-множественной операции отсечения многосвязной рискованной области P^1 , ограниченной множеством жордановых контуров $K^1 = \{K_1^1, K_2^1, \dots, K_n^1\}$ и односвязной рискованной области P^2 , ограниченной жордановым контуром K^2 , информационно достаточной по растровому способу построения является логико-алгебраическая модель вида:

$$K^3 = (K_1^2 \setminus K_1^1) \cup \left(\bigcup_{K_i^1 \in \Gamma^1(K_1^2)} K_i^1 \right). \quad (13)$$

Отметим, что представленные логико-алгебраические модели (10 ÷ 13) базируются на соотношениях (2 ÷ 9) в контурном представлении и для них также справедливы соотношения (9).

Заключение

Представленные результаты и теоретические обоснования по построению специального метода синтеза логико-алгебраических моделей рискованных полей проекта позволяют расширить научно-методический базис [4-14] по разработке математического обеспечения современных систем растровой машинной графики для этапа анализа и идентификации рисков проекта экспертным способом [6,15].

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Таганов А.И. Методы представления сложной структурно-символьной информации // Научно-технический журнал «Вестник РГРТА», 2006. Вып. 18. С. 74 -80.
2. ANSI/PMI 99-001-2004. Руководство к Своду знаний по управлению проектами (Руководство РМВОК). - 388 с.
3. Таганов А.И. Построение изображений физических полей на основе декомпозиции контурной информации: Межвуз. сборник научных трудов «Методы и приборы контроля параметров биосферы. - Л.: ЛИАП, 1984. Вып. 171. С. 90-94.
4. Таганов А.И. Теоретико-множественные операции геометрического моделирования контурных изображений// Сборник научных трудов «Автоматизация проектирования микроэлектронных вычислительных средств». Рязань: РГРТУ, 1990. С. 85- 91.
5. Таганов А.И. Применение нечетких множеств для формализации процессов анализа и идентификации важности рисков программного проекта // Научно-технический журнал «Системы управления и информационные технологии», 2007. № 4(30). С. 46-51.
6. Корячко В.П., Таганов А.И. Программный метод управления рисками качества проекта информационной системы // Научно-технический журнал «Известия Белорусской инженерной академии». Выпуск 1(17)/4, 2004. - С. 168-179.
4. Формальные методы поддержки процесса управления рисками качества проекта: учеб. пособие / А.И.Таганов, Р.А.Таганов; под ред. В.П.Корячко. Рязань: РГРТА, 2003. - 76 с.

8. Таганов А.И., Таганов Р.А. Методологические основы методов идентификации рисков событий проекта // Научно-технический журнал «Вестник РГРТА». Рязань: РГРТА, 2003. Вып. 12. - С. 70-77.

9. Таганов А.И. Применение нечетких множеств для формализации процессов анализа и идентификации важности рисков программного проекта // Научно-технический журнал «Системы управления и информационные технологии». - Москва-Воронеж. Выпуск №4(30), 2007. - С. 46-51.

10. Таганов А.И., Таганов Р.А. Метод определения оптимальной альтернативы реагирования на этапе мониторинга рисков проекта // Научно-технический журнал «Вестник РГРТА». - Рязань: РГРТА, 2003. Вып. 11. - С. 115-118.

11. Таганов А.И., Таганов Р.А. Применение нечетких ситуационных моделей для идентификации рисков программного проекта // Научно-технический журнал «Системы управления и информационные технологии». - Москва-Воронеж. № 4.2(30), 2007. - С. 297- 303.

15. Таганов А.И., Таганов Р.А. Разработка инструментальных средств поддержки процессов управления рисками качества программного проекта // Научно-технический журнал «Вестник РГРТА». - Рязань: РГРТА, 2003. Вып. 13. С. 52-57.

С.Н. ФОМИН, Д.А. ПЕРЕПЕЛКИН

Рязанский государственный радиотехнический университет

РАЗРАБОТКА КОМПЛЕКСА ПРОГРАММНЫХ СРЕДСТВ МОДЕЛИРОВАНИЯ АЛГОРИТМОВ МАРШРУТИЗАЦИИ В КОРПОРАТИВНЫХ СЕТЯХ

Разработан пакет прикладных программ моделирования алгоритмов маршрутизации, позволяющий оценить эффективность функционирования корпоративных сетей на сетевом уровне.

Введение. Современные корпоративные вычислительные сети (КВС) вышли на качественно новый уровень своего развития. Для него характерно резкое возрастание количества взаимодействующих узлов, интенсивность обмена данными, активное использование мультимедиа-технологий в производственном процессе, повышенные требования к оперативности доставки информации. Сильная территориальная разобщенность таких сетей с одной стороны, и динамический характер параметров сетевых компонентов и топологии с другой, подразумевает

создание и использование качественно новых подходов к управлению передачей данных [1].

Постановка задачи. Одной из главных задач современных корпоративных сетей является улучшение динамики ее работы, а именно выбор эффективного алгоритма маршрутизации, который будет обеспечивать поиск оптимальных маршрутов передачи данных в корпоративных сетях.

Для количественной оценки эффективности функционирования корпоративных сетей разработан пакет прикладных программ имитационного моделирования алгоритмов маршрутизации, которые применяются в настоящее время для построения таблиц маршрутизации в корпоративных сетях.

На данный момент существуют программные продукты позволяющие моделировать корпоративные сети с более детальной настройкой каждого узла сети. Однако они достаточно дорого стоят, более сложны в эксплуатации, требуют глубоких настроек и практических навыков в работе с корпоративными вычислительными сетями.

В связи с этим стоит задача в разработке новой программной системы с более удобным и простым графическим интерфейсом, позволяющей оценить эффективность работы алгоритмов маршрутизации в корпоративных вычислительных сетях.

Выбор среды разработки. Средой разработки проекта является Borland Delphi 7.0. Язык программирования - Object Pascal. Такой выбор обусловлен тем, что Borland Delphi является средой визуального объектно-ориентированного программирования, это позволяет создавать с небольшими затратами сил и времени программные продукты. Delphi – это многозадачная среда, позволяющая реализовывать все необходимые операции в программе и легко читаемый интерфейс для удобной работы пользователя с программой.

Описание продукта. Предложенный пакет программ может применяться в учебных целях для оценки эффективности алгоритмов маршрутизации в корпоративных сетях.

В разработанном пакете программ корпоративная сеть представлена в виде неориентированного взвешенного графа, вершинами которого являются маршрутизаторы, а ребрами – линии связи между ними.

Каждая линия связи характеризуется своим весом. Вес линии связи между маршрутизаторами можно представить для каждого алгоритма по-разному, так как реализованные в предложенном пакете программ алгоритмы маршрутизации, используются в различных протоколах маршрутизации.

Укрупненная схема работы пакета программ представлена на рис. 1:

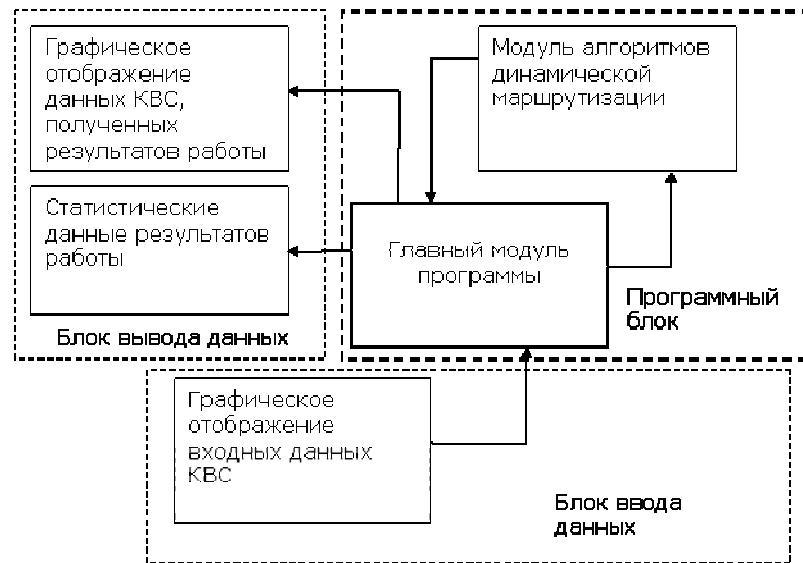


Рис.1. Схема работы пакета программ

В разработанном пакете программ реализована возможность определять оптимальные маршруты передачи данных от узла – источника до всех остальных узлов сети с помощью алгоритмов поиска оптимальных путей в графах с фиксированной структурой, производить построение таблиц маршрутизации необходимого участка сети. Кроме того, для базовых алгоритмов маршрутизации реализована возможность пошагового отображения их работы.

Главное окно программы моделирования алгоритмов маршрутизации в корпоративных сетях показано на рис. 2.

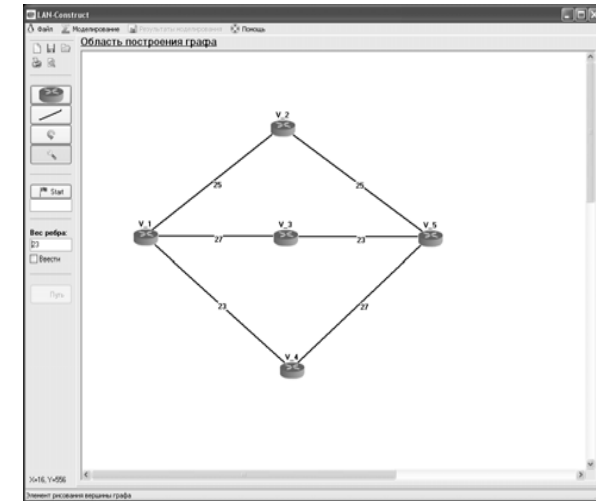


Рис. 2. Главное окно пакета программ

Результаты работы предложенного пакета программ могут быть представлены в виде таблицы маршрутизации, в пошаговом режиме и наглядно на структуре корпоративной сети путем выделения другим цветом оптимальных маршрутов.

Заключение. Разработанный пакет программ моделирования алгоритмов маршрутизации позволяет определить оптимальные маршруты передачи данных и оценить эффективность современных алгоритмов маршрутизации в корпоративных сетях с различной структурой. Практическое использование предложенного пакета программ в учебном процессе позволяет наглядно продемонстрировать работу корпоративных сетей и применяемых в них алгоритмов на сетевом уровне.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Олифер В.Г., Олифер Н.А. Новые технологии и оборудование IP-сетей. – СПб.: БХВ-Санкт-Петербург, 2001. – 512 с.

А.С. ХАРЧЕНКО, Ю.М. МИНАЕВ, В.Н. РУЧКИН

Рязанский государственный университет им. С. А. Есенина

СИСТЕМА УПРАВЛЕНИЯ ИСТОРИЧЕСКИМИ ЗНАНИЯМИ

В работе рассматриваются проблемы создания систем управления историческими знаниями и формализации исторических знаний.

В современном информационном обществе одним из наиболее перспективных и развивающихся направлений является создание баз знаний и экспертных систем, массовое использование которых должно привести к созданию единого образовательного информационного пространства. Известно [1], что наиболее важным параметром БЗ является качество содержащихся знаний, достоверность конкретных и обобщенных сведений в базе данных и релевантности информации.

Однако в процессе проектирования ЭС в области исторических знаний возникают следующие проблемы:

Сложность формализации гуманитарных знаний.

Недостаточное исследование исторических аспектов систем управления историческими знаниями

Имеются проблемы глубокого исследования данных систем в конкретных исторических фактах

Выбранный исторический период «Холодной войны» носит актуальный характер в современных условиях пересмотра исторических событий.

Разработанная экспертная система имеет ряд преимуществ перед человеком-экспертом в предложенной области и имеет следующие достоинства:

□ Повышенная доступность, которая обеспечивается за счет возможности применения любых подходящих компьютерных аппаратных средств.

□ Уменьшенные издержки за счет снижения стоимости предоставления экспертных знаний в расчете на отдельного пользователя.

□ Возможность получения вложенных экспертных знаний, путем объединения знаний нескольких экспертов, превышающих уровень знаний отдельно взятого эксперта-человека.

□ Экспертная система может действовать в качестве интеллектуальной обучающей программы, передавая учащемуся на выполнение примеры программ и объясняя, на чем основаны рассуждения системы.

Актуальность периода «Холодной войны» объясняется противостоянием центральной Европы и России, обострившейся в связи с распадом СССР и Организации Варшавского Договора.

На первоначальном этапе создания системы необходимо было решить важную проблему формализации знаний исторического характера. Данная задача была решена путем выделения наиболее значительных сведений об историческом событии и их объединение во фрейм.

Табл. 1. Пример формализованного события

| Дата | Событие | Экономическое влияние | Тактическое значение | Стратегическое значение |
|--------------|--------------------------|-------------------------------------------------------------|-----------------------------------------|-----------------------------------------------------------------|
| 5 марта 1946 | Фултонская речь Черчилля | Прекращение экономических связей между соц. и кап. лагерями | Провозглашение открытого противостояния | Возникновение «железного занавеса», разделение мира на 2 лагеря |

Для создания информационной системы был использован логический язык программирования Visual Prolog, с помощью которого созданы графический интерфейс пользователя и база знаний. При реализации пользовательского интерфейса экспертной системы был реализован принцип «делай то, что я имею в виду» или DWIM (англ. Do What I Mean). DWIM требует, чтобы система работала предсказуемо, чтобы пользователь заранее интуитивно понимал, какое действие выполнит программа после получения его команды. Этот фактор является очень важным при построении экспертных систем, т.к. от удобства и понятности интерфейса зависит скорость и точность запроса к базе знаний.

Основной рабочей областью системы является диалоговое окно – окно программы, в котором производится формирование запроса к базе знаний и вывод результатов. Запросы формируются с помощью выпадающих меню, результат выводится в специальное поле. В данной системе реализована возможность поиска решений по неполным введенным данным.

В данной работе, представляющей экспертную систему по исторической тематике «Холодной войны», выполнены следующие требования:

1. Возможность вербализации задачи
2. Способность иметь дело с неопределенностью и неполнотой
3. Возможность простой актуализации предметной области

4. Возможность безболезненного расширения охвата знаний внутри предметной области и работы на нескольких языках.

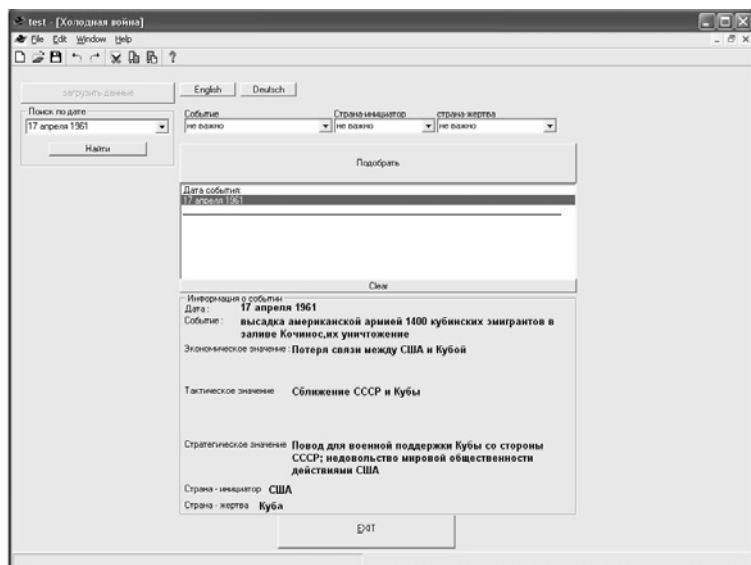


Рис. 1. Интерфейс системы

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Искусственный интеллект: современный подход. 2 издание/ Стюарт Рассел, Питер Норвиг. -М; «Вильямс»2006.-1048с.
2. Основы искусственного интеллекта/Б.В.Костров, В.Н. Ручкин, В.А. Фулин.-М.: «ДЕСС», «ТехБук», 2007.-222с.
3. Универсальный искусственный интеллект/В.Н. Ручкин, В.А. Фулин.-С.-Пт.: «БХВ-Петербург», 2009.-30с

Р.В. ХРУНИЧЕВ

Рязанский государственный радиотехнический университет

ЭЛЕКТРОННЫЕ ИЗДАНИЯ: ТЕРМИНОЛОГИЯ И КЛАССИФИКАЦИЯ

Разрабатывается терминология и классификация электронных изданий, которые, в последствие, внедрены в учебный процесс РГРТУ.

При внедрении системы дистанционного обучения в любом вузе возникает необходимость разработки терминологии, а также создания классификации электронных изданий образовательного назначения (ЭОИ). Эта задача является, несомненно, актуальной, так как она является одной из самых главных и самой первой при создании любых электронных изданий.

Наш вуз не стал исключением, преподаватели при разработке собственных курсов столкнулись с проблемой терминологии и классификации ЭОИ. Один из возможных вариантов используемой терминологии и классификации ЭОИ представлен ниже.

Информационные ресурсы – отдельные документы и отдельные массивы документов, документы и массивы документов в информационных системах (архивах, фондах, банках данных и т. д.) [3].

Контент сам по себе – не есть средство обучения. Контент – это то, из чего создаются электронные издания (ЭИ), собственно, и являющиеся информационными ресурсами, это совокупность: графической, цифровой, речевой и музыкальной информации, текста, видео и фото.

Электронное учебное (образовательное) издание – ЭИ, предназначенное для автоматизации обучения и контроля знаний, и соответствующее учебной дисциплине или отдельным ее частям, а также позволяющее определить траекторию обучения, содержащее систематизированный материал по соответствующей научно-практической области знаний, обеспечивающее творческое активное овладение студентами **знаний, умений и навыков** в этой области [3].

Электронное учебное (образовательное) издание (ЭУ(О)И), содержащее систематическое изложение учебной дисциплины или её раздела и обладающее официальным статусом данного вида издания – **электронный учебник (ЭУ)** [3].

Более широким понятием является электронное издание (ЭИ), состав которого представлен на рис. 1.



Рис. 1. Состав ЭИ

Классификация электронных образовательных изданий – одна из актуальных задач. Ниже приведено несколько существующих классификаций, не единственных (рис. 2). Это говорит о том, что нет однозначного подхода к образовательным изданиям, идет непрерывный поиск оптимального варианта.



Рис. 2. Вариант классификации ЭУ(О)И

Применительно к системе обучения в целом, электронные учебные (образовательные) издания можно классифицировать следующим образом:

- 1) используемые в традиционной системе в соответствии с ГОСО;
- 2) энциклопедического характера;
- 3) домашние репетиторы;
- 4) справочного характера;
- 5) контролирующие и оценивающие результаты учебной деятельности;
- 6) для факультативной работы.

В традиционной системе обучения ЭУ(О)И – это все типы программных средств. ЭУ(О)И могут подразделяться в зависимости от формы организации занятия, отличаться по форме изложения материала, различаться по характеру размещения на носителях – однотомные, многотомные и электронные сессии.

По технологии распространения ЭУ(О)И могут быть локальными, комбинированными и сетевыми.

Встречаются неявные способы классификации, в частности, по стратегии обучения, по природе использования правил ввода и вывода предоставляемого учебного материала.

Различные виды ЭУ(О)И и материалы для их разработки могут делиться на несколько групп, например, издания носят характер исходного материала, из которого впоследствии разрабатываются полноценные ЭУИ.

Более понятна классификация по методическому назначению:

- | | |
|-----------------------------|------------------------------|
| 1) учебно-игровые; | 2) игровые; |
| 3) тренажеры; | 4) лабораторные; |
| 5) информационно-поисковые; | 6) моделирующие; |
| 7) обучающие; | 8) имитационные; |
| 9) контролирующие; | 10) демонстрационные; |
| 11) расчетные; | 12) информационно-справочные |

Ресурсы определяются в стандарте как учебно-методические материалы по курсу, дисциплине. Лицо, отвечающее за составление контента информационного наполнения дисциплины, называется разработчиком дисциплины.

Совокупность программ, с помощью которых можно реализовать цели и задачи дистанционного обучения и обеспечить активную учебную деятельность, называется программным обеспечением системы ДО.

Получается, что контент и ресурсы напрямую связаны друг с другом. Контент служит материалом создания средств обучения. Средства обучения – это одна из форм представления содержания обучения. Средства обучения в своем большинстве реализуются через средства компьютерных и телекоммуникационных технологий. На преподавателей-разработчиков возлагается задача создания ресурса.

Поставленная задача была новой для нашего вуза, т.к. система дистанционного обучения только внедряется в учебный процесс. Для создателей курсов мной была предложена классификация ЭОИ, которая, на мой взгляд, более полно отражает специфику дистанционного обучения. Предложенная классификация используется при разработке учебных курсов в системе ДО Moodle и на подготовительных курсах повышения квалификации преподавателей РГРТУ.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Шакаримова А.Б., Криулько Н.С. Дистанционное обучение (опыт реализации в ВКГТУ). – Усть-Каменогорск: ВКГТУ, 2006. – 107с.
2. Белозубов А.В., Николаев Д.Г. Система дистанционного обучения Moodle. – Спб., 2007. – 108 с.
3. ФЗ от 20 февраля 1995 г. № 24-ФЗ «Об информации, информатизации и защите информации».