

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

ЮЖНО-РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
(НОВОЧЕРКАССКИЙ ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ)

**ТЕОРИЯ, МЕТОДЫ ПРОЕКТИРОВАНИЯ,
ПРОГРАММНО-ТЕХНИЧЕСКАЯ ПЛАТФОРМА
КОРПОРАТИВНЫХ
ИНФОРМАЦИОННЫХ СИСТЕМ**

*Материалы
X Международной научно-практической
конференции*

*5 июня 2012 года,
г. Новочеркасск*

Новочеркасск
ЮРГТУ
2012

УДК 004
ББК 32.973
Т 33

Организаторы конференции:

Министерство образования и науки РФ;
Южно-Российский государственный технический университет
(Новочеркасский политехнический институт);
Южно-Российский региональный Центр информатизации высшей
школы (ЮГИНФО);
Южный федеральный университет;
Технический университет г. Ильменау (ФРГ);
Северо-Кавказский государственный технический университет,
г. Ставрополь;
Ростовский государственный экономический университет «РИНХ»;
Анапский филиал Российского государственного социального
университета;
Инвестиционная компания «Фондовый дом «Сельмашинвест»;
Институт управления, бизнеса и права (г. Ростов-на-Дону);
Корпорация «Галактика» (г. Москва);
Донской филиал Центра тренажеростроения (г. Новочеркасск);
Институт открытого образования (г. Шахты);

Оргкомитет конференции:

Н.И. Горбатенко (РФ), профессор – председатель;	А.В. Чернов (РФ), профессор;
Л.А. Крукиер (РФ), профессор;	И.В. Дарда (РФ), доцент;
Ю.И. Рогозов (РФ), профессор;	С.П. Воробьев (РФ), доцент – зам. председателя;
В.Ф. Гузик (РФ), профессор;	А.Г. Душенко (РФ), доцент;
Н.В. Долматова (РФ), профессор;	Ю.А. Бахвалов (РФ), профессор;
Ю.В. Дашко (РФ), профессор;	А.В. Павленко (РФ), профессор;
Е. Калленбах (ФРГ), профессор;	А.М. Безуглов (РФ), доцент;
В.И. Дроздова (РФ), профессор;	Н.П. Вычужанина (РФ), доцент
Г.Н. Хубаев (РФ), профессор;	

Редакционная коллегия:

Н.И. Горбатенко, профессор – ответственный редактор;
С.П. Воробьев, доцент – зам. ответственного редактора

Т 33 **Теория, методы проектирования, программно-техническая платформа корпоративных информационных систем:** материалы IX Междунар. науч.-практ. конф., г. Новочеркасск, 5 июня 2012 г. / Юж.-Рос. гос. техн. ун-т (НПИ). – Новочеркасск: ЮРГТУ, 2012. – 178 с.

ISBN 978-5-9997-0268-5

Материалы конференции посвящены архитектурным и системным решениям для корпоративных информационных систем, тренажерных и обучающих комплексов, оптимальному проектированию программно-технической платформы, а также методам и моделям оценки их производительности и масштабируемости.

УДК 004
ББК 32.973

ISBN 978-5-9997-0268-5

© Южно-Российский государственный
технический университет (НПИ), 2012

УДК 004.09:331.108.66:334.012.42

ИНФОРМАЦИОННО-АНАЛИТИЧЕСКАЯ СИСТЕМА ОТДЕЛА КАДРОВ КОММЕРЧЕСКОГО ПРЕДПРИЯТИЯ

Вороненко А.В., Щербакова Е.А.

Южно-Российский государственный технический университет (НПИ), г. Новочеркасск

Описывается информационно-аналитическая система отдела кадров. Приведены возможности и преимущества использования для предприятия с позиции эффективности и экономии.

Управление кадрами является индивидуальным процессом для каждой организации, поскольку хозяйственная деятельность каждой организации имеет свою отличительную регламентирующую внутреннюю документацию. В зависимости от формы оплаты труда, особенностей анализа эффективности использования трудовых ресурсов и учёта рабочего времени процесс управления кадрами может быть разным.

На сегодняшний день ООО «ИСН–7» занимает лидирующие позиции среди множества IT-компаний данного направления в России и в частности по Ростовской области, постоянно прогрессируя и увеличивая штат своих сотрудников. Поэтому важнейшим фактором стабильной работы ООО «ИСН–7» является эффективное управление кадрами, что позволяет снизить текучесть кадров, увеличить штат и профессионализм сотрудников, повысить прибыль компании, сократить время простоя незакрытых вакансий, а также сократить временные и финансовые затраты на подбор персонала.

Предлагаемая информационно-аналитическая система управления кадрами ООО «ИСН–7» соответствует ТК РФ и позволяет автоматизировать следующие виды работ сотрудников отдела кадров ООО «ИСН–7»:

- учёт рабочего времени IT-сотрудников;
- анализ временных трудовых затрат и оценка эффективности использования человеческих ресурсов;
- определение необходимости привлечения человеческих ресурсов;
- оптимизация процессов рекрутинга на вакантные должности;
- создание системы отчетности по показателям деятельности ОК и организации в целом;
- ведение регламентированного документооборота;
- отражение внутренних отчетов и документов организации.

Структурированное описание системы управления кадрами ООО «ИСН–7» и её взаимодействие с внешней средой представлено ниже диаграммой декомпозиции на рис. 1.

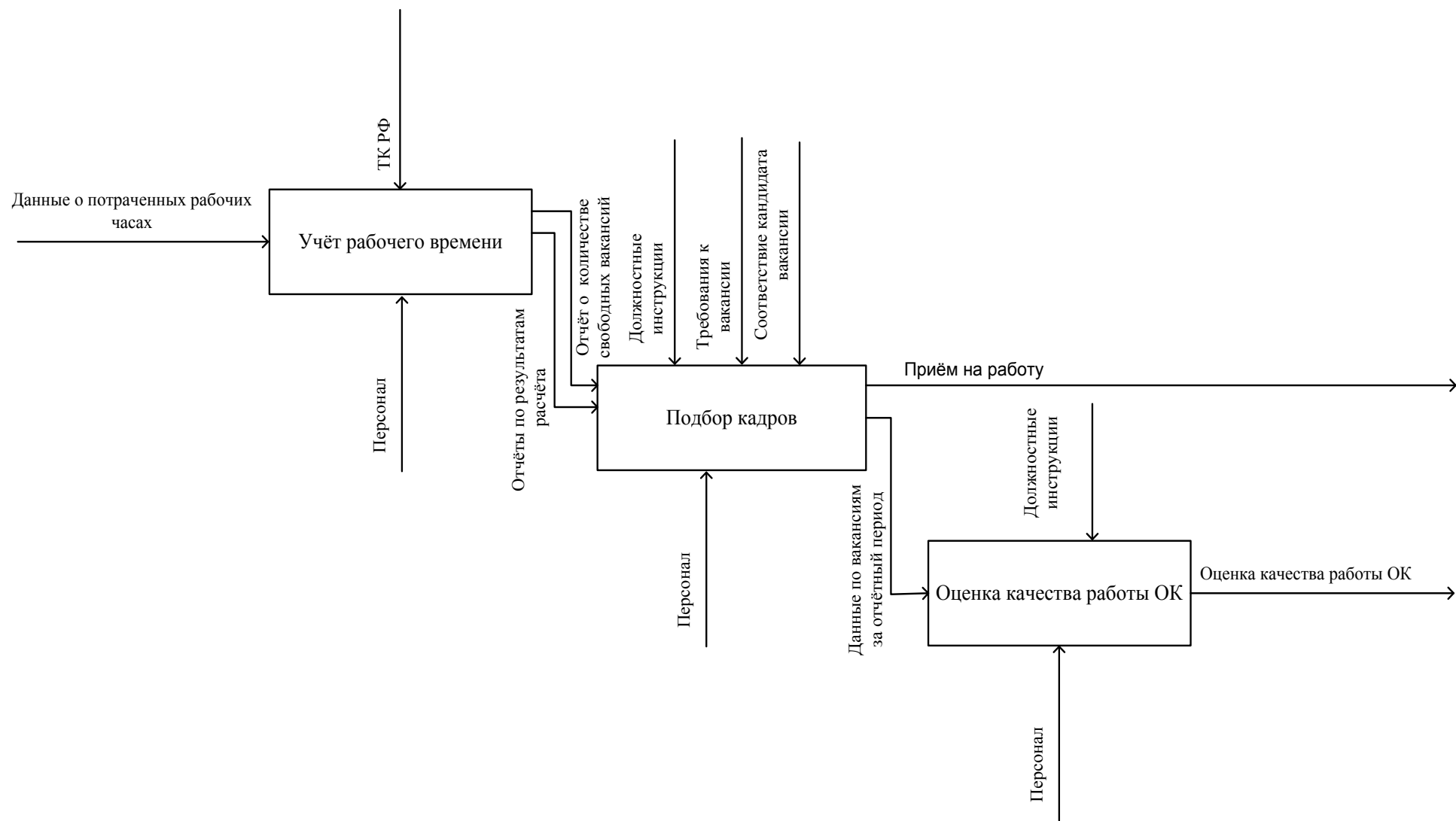


Рис. 1. Диаграмма декомпозиции управления кадрами

Предлагаемая информационно-аналитическая система способствует достижению следующих целей:

- сокращение времени, затрачиваемого на выполнение должностных сотрудниками отдела кадров;
- повышение качества работы отдела кадров за счет оперативности представления, полноты, достоверности и удобства форматов отображения информации и сокращение времени на принятие решения;
- повышение эффективности исполнения процессов, путем сокращения непроизводительных и дублирующих операций, выполняемых «вручную».

Поставленная задача реализована на базе технологической платформы «1С Предприятие 8.2», на основе типовой конфигурации «1С: Зарплата и управление персоналом 8».

УДК 004.522

АЛГОРИТМ ПОИСКА НАИЛУЧШЕЙ ОБЛАСТИ ОПРЕДЕЛЕНИЯ ПРИ ОПТИМАЛЬНОМ СИНТЕЗЕ И РАСПОЗНАВАНИИ РЕЧЕВЫХ ОБРАЗОВ

Гавриков М.М., Синецкий Р.М.

Южно-Российский государственный технический университет (НПИ), г. Новочеркасск

Представлены результаты экспериментальной оценки эффективности алгоритма поиска наилучшей области определения в схеме структурно-аппроксимационного синтеза и распознавания структурных речевых образов, рассмотрены несколько примеров использования алгоритма в указанных задачах.

В настоящее время для управления информационными процессами, протекающими в различных системах, включая тренажерные системы, все большее применение находят речевые интерфейсы и командно-речевые интерпретаторы (КРИ). В ряде предшествующих публикаций для реализации КРИ, используемого как библиотека речевого управления, была разработана математическая схема структурно-аппроксимационного синтеза (СА-схема) и распознавания структурных речевых образов, формируемых из речевых сигналов [1, 2]. Разработанный КРИ может использоваться в двух режимах: режиме «постоянно активного микрофона» и режиме «активации микрофона при помощи кнопки», используемого, например, в мобильных телефонах. Особенность и сложность реализации режима «постоянно активного микрофона» заключается в том, что кроме речевых команд

управления микрофон улавливает и все посторонние акустические шумы (фрагменты посторонней речи, акустические шумы, близкие к речевому диапазону частот и т. п.), и соответствующие шумовые сигналы, в принципе, могут быть распознаны как команды управления. Чтобы свести к минимуму такие ситуации КРИ должен иметь механизм распознавания «ложных» команд. В КРИ, использующем первую версию СА-схемы, этот механизм был реализован следующим образом.

При поступлении в микрофон акустического сигнала КРИ выполняет распознавание речевой команды и передает код этой команды исполнительной процедуре или интерпретирует ее как «ложную». Блок распознавания для определения индекса r поданной команды среди L возможных команд и ее интерпретации использует правило

$$r = \arg \min_{1 \leq i \leq L} \{\Omega_i^*\} \text{ при условии что} \\ \Omega_r^* < P_r, \quad (1)$$

где Ω_i^* – значение структурно-аппроксимационного критерия, соответствующее оптимальному аппроксимационному речевому образу для i – й команды, P_r – допустимое пороговое значение критерия при правильном распознавании r – й команды.

Пороговые значения $P_i, (i = \overline{1, L})$ входят в состав характеристик идеальных (эталонных) речевых образов и определяются заранее на этапе настройки КРИ на заданный список речевых команд. Если условие (1) не выполняется, то команда отвергается (игнорируется) или выдается голосовое сообщение с просьбой повторить команду. Таким образом, результат распознавания и интерпретации формально определяется значениями двух пар – $(r, 0)$ и $(r, 1)$, имеющих следующую семантическую интерпретацию:

$(r, 0)$ – "команда, отвергнуть"; $(r, 1)$ – "команда, выполнять".

Мониторинг результатов экспериментов в режиме «постоянно активного микрофона» показал, что среди множества команд с результатом интерпретации $(r, 0)$ значительную долю могут составлять правильно распознанные команды (с правильным значением кода r), но отвергнутых как «ложные» по условию (1). В большинстве случаев это происходит из-за погрешности в определении границ речевого сигнала: захватывании участков фонового шума микрофона, участков, соответствующих фрагментам посторонней речи и акустическим помехам в речевом диапазоне. Вследствие этого, значение критерия Ω_i^* оказывается большим, чем то, которое

могло бы быть в отсутствии указанных факторов (условие (1) не выполняется), и команда отвергается. Такие ситуации требуют повторного, иногда многократного произнесения команд, что снижает эффективность речевого управления. Для уменьшения влияния акустических помех на результат распознавания была разработана модифицированная СА-схема, использующая алгоритм поиска наилучшей области определения (алгоритм ПНОО) для речевого образа. Применение новой версии СА-схемы позволяет значительно снизить долю отвергнутых команд при возникновении описанных выше ситуаций. Это достигается за счет того, что алгоритму ПНОО удастся найти лучшую по сравнению с первоначальной область определения речевого образа, на которой значение критерия Ω_i^* уменьшается и удовлетворяет условию (1).

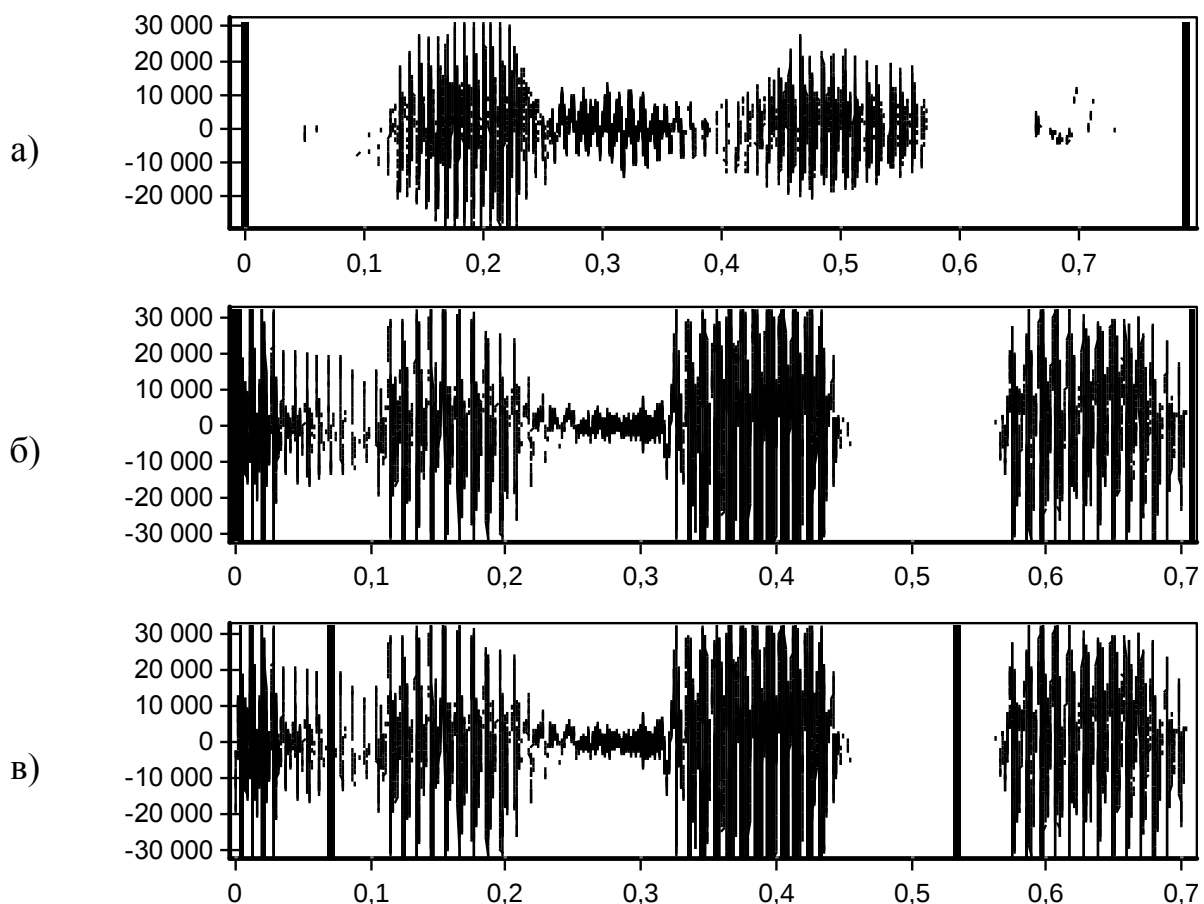


Рис. 1. Иллюстрация применения алгоритма на примере команды «назад»

Вышеизложенное иллюстрируют графики результатов экспериментов, показанные на рис. 1. На рис. 1,а показан речевой сигнал, соответствующий эталонному образцу команды «назад». На рис. 1,б показан сигнал, выделенный КРИ после подачи этой команды оператором. Как видно, при сравнении с эталоном этот выделенный сигнал содержит фрагменты

фоновых помех микрофона в начале и в конце записи. Результат его распознавания и интерпретации – «назад, отвергнуть», так как для оптимального образа, соответствующего этому сигналу, условие (1) не выполняется. На рис 1,в показан тот же сигнал, но с областью определения (ее границы обозначены вертикальными линиями), полученной после «подгонки» при помощи алгоритма ПНОО. Результат его распознавания и интерпретации – «назад, выполнить», так как после «подгонки» для оптимального образа, соответствующего этому сигналу, условие (1) выполняется.

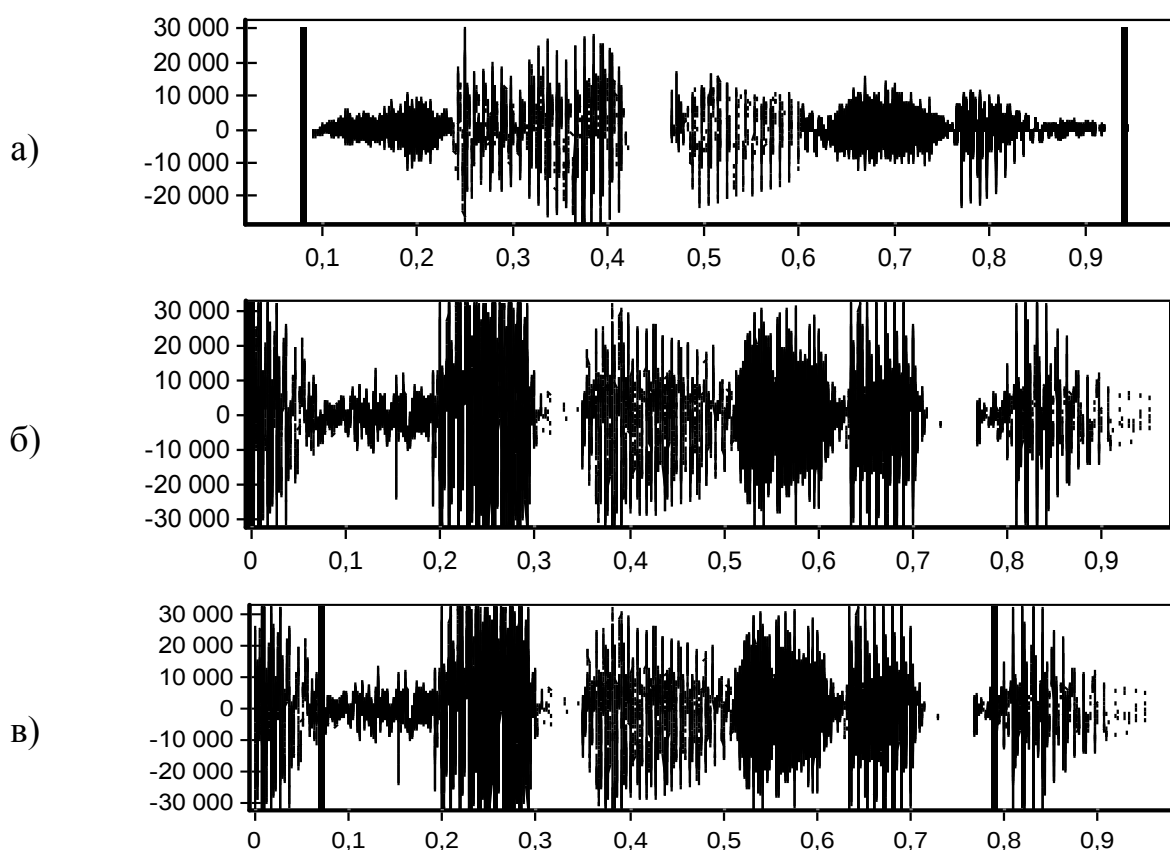


Рис. 2. Иллюстрация применения алгоритма на примере команды «следующий»

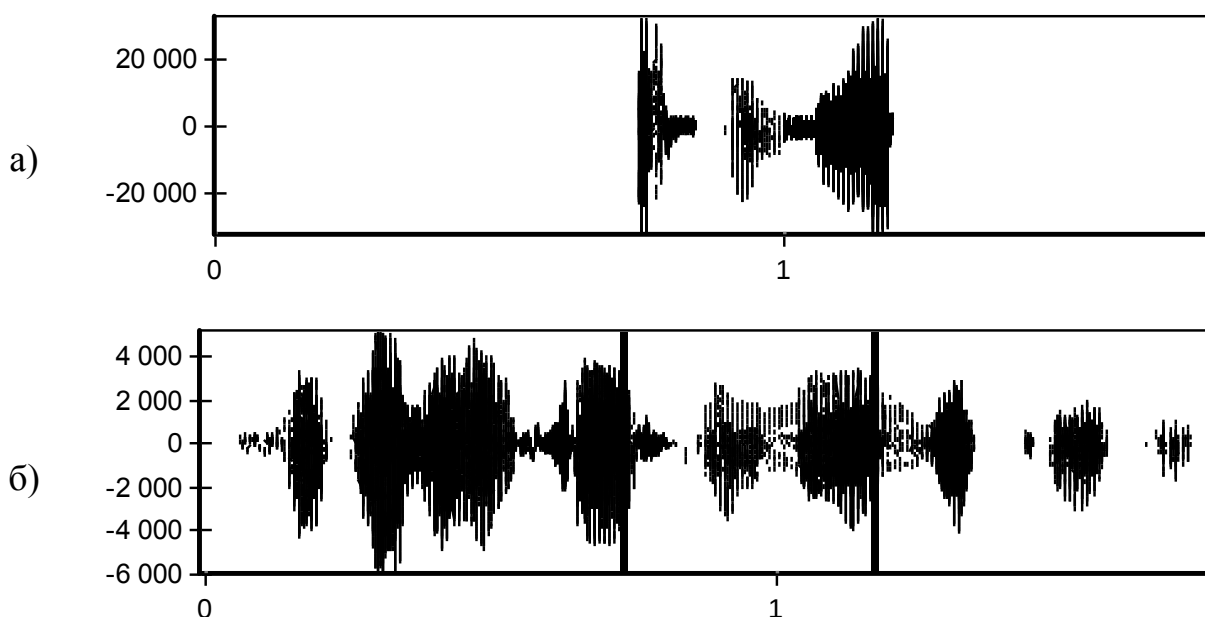
Аналогичный пример показан на рис. 2 для команды «следующий», когда в начале выделенного речевого сигнала (рис. 2,б) содержится отрезок, соответствующий фрагменту посторонней речи, воспринятой микрофоном. Сводные результаты экспериментов приведены в таблице 1.

Таблица 1

Кол-во поданных команд	Количество правильно определенных кодов команд r	Количество «отвергнутых» команд по превышению порога	Количество «возвращенных» после подгонки области определения
100	93	45	36

В условиях специально создаваемых сильных акустических помех подавалось 5 различных команд по 20 раз каждая. Как видно из таблицы, почти половина команд была отвергнута по превышению порога, и в этих случаях пришлось бы повторно произносить практически каждую вторую команду. Применение алгоритма ПНОО позволило значительно сократить количество отвергнутых команд до 9 команд ($45 - 36 = 9$). Общий процент правильно интерпретируемых команд, а, следовательно, не требующих повторного произнесения, составил 84 %. Следует учесть, что это относится к случаю, когда сильные помехи создаются искусственно. Поэтому следует ожидать, что в реальных условиях это процент будет выше.

Другое применение алгоритма ПНОО может заключаться в автоматическом поиске фрагментов речевого сигнала, соответствующих ключевым словам/фразам в речевых файлах большого объема, например в записях переговоров диспетчеров электросетей, диспетчеров РЖД, авиационных диспетчеров, «черных ящиках» и т.д. Это применение иллюстрирует рис. 3.



**Рис. 3. Иллюстрация применения алгоритма ПНОО
для поиска фрагментов речевого сигнала**

На рис. 3,а показана запись речевого сигнала, соответствующего слову «основной», которое должно быть найдено, а на рис. 3,б – запись речевого сигнала, соответствующего фразе «Загорелся основной индикатор» с найденной при помощи алгоритма ПНОО областью определения оптимального речевого образа искомого слова «основной».

Как видно из рисунков, нужный фрагмент сигнала выделен достаточно точно, хотя в данном приложении большая точность, по всей видимости, и не требуется.

Литература

1. Гавриков М.М. Структурная аппроксимация и распознавание одномерных временных образов. Концепция и применения. // Изв.вузов. Электромеханика. – 2003. – №6.
2. Гавриков М.М., Синецкий Р.М. Алгоритмическая и численная реализация структурно-аппроксимационного метода распознавания речевых образов. // Изв.вузов. Электромеханика. – 2007. – №2. – С. 52-59.

УДК 004.9:378.225(47+57)

СРАВНИТЕЛЬНЫЙ АНАЛИЗ ПОДХОДОВ К ПОДГОТОВКЕ ИТ-СПЕЦИАЛИСТОВ

Горбатова Ю.А.

Южно-Российский государственный технический университет (НПИ), г. Новочеркасск

Рассмотрены отдельные подходы и методики подготовки ИТ-специалистов в учебных центрах известных компании Softline, Cisco, Cleverics, Careerlab, Microsoft, а также в российских вузах, таких как Московский государственный университет имени М. В. Ломоносова и Международный Университет «Дубна»

Сегодня наступила эпоха глобальной информатизации. Информационные технологии стали одной из основных отраслей экономики, обеспечивающей высокотехнологичные инновационные производства. Все чаще возникают проблемы с нехваткой высокопрофессиональных специалистов в ИТ-сфере из-за постоянной новизны и стремительного развития данной области знаний, разнообразия квалификационных требований к подготавливаемым специалистам. Тема подготовки ИТ-специалистов стала одной из главных в вопросах о перспективах ее развития. Представители компьютерного бизнеса в последние годы все чаще затрагивают ее в диалоге с государством, обращая внимание на то, что главным поставщиком кадров выступает высшая школа. В связи с этим, все настойчивее поднимаются вопросы об изменении методик и программ подготовки студентов по ИТ-специальностям и их адаптации к требованиям времени, способностям реализовывать ИТ-проекты различной сложности и масштабов, а также увеличении числа выпускников по этим направлениям.

По данному вопросу своим мнением делятся эксперты учебных центров по обучению ИТ-специалистов во время проведения виртуального

круглого стола, подчеркивая особенности профессионального обучения и проблемы, связанные с ними, а так же способы и методы, применяемые учебными центрами для их решения. Своей точкой зрения делится инструктор Учебного центра компании Softline Максим Фомин, где особенностью обучения ИТ-специалистов он видит в получение необходимых знаний в смежных областях, которые имеют непосредственное отношение к процессу обучения. Поскольку в настоящее время ИТ – это единая, неделимая сфера управления потоками информации – необходимо учитывать тот факт, что невозможно рассматривать данный сегмент в отрыве от других сфер информационных комплексов и систем. Понимание этих зависимостей и причинно-следственных связей является ключевым знанием для любого ИТ-специалиста, который хочет стать профессионалом в своей области.[1] Ведущая международная компания Softline в сфере лицензирования программного обеспечения и оказания полного спектра ИТ-услуг по обучению, консалтингу, технической и юридической поддержке, ИТ-аутсорсингу является одним из лидеров среди авторизированных учебных центров в России. Одновременно с теоретическим курсом обучения в авторизованном центре Softline для практических работ используются специальные удаленные стенды и системы виртуализации вычислительных систем, развернутые локально. Благодаря практическим знаниям и доступу к самым передовым технологиям посредством стендов слушатели получают необходимые навыки и на практике могут проверить те или иные режимы работы информационных систем. Доступ к стендам и системам виртуализации доступен только на авторизированных курсах Учебного центра Softline. После окончания обучения слушателям выдаются международные дипломы от вендора и дипломы от Учебного центра Softline, которые признаются в любой стране мира без каких-либо дополнительных тестирований или подтверждений уровня квалификации слушателя.

Директор департамента обучения компании Cleverics Роман Журавлев выделяет такие особенности в ИТ-индустрии, как стремительное развитие ИТ-отрасли и потребность в постоянном обучении в этой сфере; постоянное обновление технической экспертизы; нехватка менеджерских навыков и знаний об управлении информационными технологиями у российских ИТ-специалистов, поэтому на курсах в Cleverics обучают именно управлению услугами, взаимодействию с заказчиками, эффективную организацию ИТ-деятельности, где активно развивается дистанционное обучение, которое может стать основной формой обучения специалистов из регионов.[1] Отличительной чертой компании Cleverics в обучении специа-

листов являются создание, разработка и распространение профессиональных бизнес-симуляций или симуляционных игр, такие как «Apollo-13 – an ITSM case experience», «The Challenge of Egypt», «The Greatest Move» и «Grab@Pizza». Они позволяющих организациям добиваться большей пользы от обучения сотрудников. В симуляционных играх подразумевается, что группа людей будет в течение дня находиться в тестовой среде, выполняя роли центра управления полётами NASA или играя роли менеджеров в игре Grab-a-Pizza. Интервью перед игрой поможет определить проблемы, на которых делается акцент в игре, а также какие вопросы будут представлены в обсуждении. В деловых играх участники могут работать над реальными вопросами, включая личные вопросы. Основной эффект таких бизнес-симуляций – это возможность использовать полученные знания в реальной жизни. [2]

В разработке ПО участвуют множество профессий: разработчики-инженеры, архитекторы, аналитики, менеджеры проектов и продуктов, специалисты по программной инженерии и по управлению программными проектами, которые, сильно отличаются, в зависимости от процессов компании и выбранной методологии разработки. Заместитель генерального директора специализированного образовательного центра Careerlab Елена Арсеньева отметила особенности обучения отдельно для каждой должности и выделила роль человеческого фактора и эффективности коммуникаций в команде разработки, так как считает, что одних блестящих технических знаний может быть недостаточно и важно учитывать соответствие программы обучения с целям и задачам компании.[2] Поэтому процесс обучения данного специализированного образовательного центра в области программной инженерии уникален тем, что кроме традиционных методов – чтения лекций, предполагается проведение практикумов и деловых игр, позволяющих отработать методы управления проектами на практике, а также выполнение тестов, которое в дальнейшем поможет развитию в качестве руководителя. Но главным отличием курсов выступает соответствие программы обучения целям и задачам компании, где обязательным условием являются актуальный опыт экспертов и инструкторов.

Для уменьшения разрыва между профилем выпускника вуза и требованиями современной ИТ-индустрии к молодым специалистам бизнес-лидеры реализуют различные модели взаимодействия с вузами. Так, компания Microsoft сотрудничает с системой высшего образования практически с начала работы московского представительства. Со многими вузами подписаны стратегические соглашения о совместном сотрудничестве по

развитию учебной программы вузов, организации специальных мероприятий, конкурсов и грантов для студентов и преподавателей, курсов и семинаров по технологиям Microsoft, и по созданию центров инноваций. Компания HP использует различные формы поддержки вузов, как совместные научные исследования, так и приема на работу студентов до клиентских и партнерских отношений. Здесь курсы технологий HP включаются в общую программу обучения соответствующего факультета и становятся доступными любому студенту с любого факультета. Сейчас у компании около 20 вузов-партнеров.

На сегодняшний день для России, как и для любой другой страны, проблема образования является одной из наиболее острых. Используя свой громадный опыт в области подготовки первоклассных IT-специалистов, компания Cisco предпринимает практические шаги в поддержку модернизации российской системы образования и повышения конкурентоспособности России. Новые программы Cisco затрагивают отечественные системы общего, среднего специального и высшего образования, и ведут переподготовку безработных и граждан в стесненных экономических обстоятельствах. К 1 июня 2011 года количество активных Сетевых академий в России достигло 217. [7]

Дефицит IT-кадров заставляет внимательно присматриваться ко всем возможным источникам кандидатов. В первую очередь это различные высшие учебные заведения, которые выпускают специалистов в области ИТ и телекоммуникаций. Лучшее образование в этих сферах дают следующие ВУЗы: МГУ им. Ломоносова, МГТУ им Баумана, МЭИ, МФТИ, МИФИ, МИРЭА, МАИ, МИЭМ, МИЭТ и «МАТИ»-РГТУ им. К. Э. Циолковского в Москве и СПбГУТ им. Бонч Бруевича в Санкт Петербурге.

Московский государственный университет имени М. В. Ломоносова является ведущим учебным центром России по подготовке IT-кадров, где учебным планом для студентов предусмотрена фундаментальная математическая подготовка. Значительное место в обучения специалистов занимает практическая работа на компьютерах, включая рабочие станции и персональные компьютеры. За время учебы студенты учатся работать в нескольких операционных системах и изучают, как минимум, три языка программирования. [3] Специфичность подготовки университетом IT-специалистов характеризуется традиционными методами, показавшими свою эффективность, в то время как Государственное образовательное учреждение высшего профессионального образования Московской области Международный Университет «Дубна» создал инновационные подходы к образовательному процессу.

Инновационное обучение вуза ориентировано на интеграцию его научно образовательного потенциала и отраслевой академической науки, установления партнерских отношений с работодателями, использование возможностей международного сотрудничества и усиление творческого, самостоятельного и заинтересованно-ответственного участия студентов в учебном процессе. Образовательная модель в университете «Дубна» использует системный подход – владение современными методами системного анализа; проблемный подход – получение навыков при решении конкретных задач из реальной практики; образный подход – применение современных графических технологий; информационный подход – использование компьютерных средств; креативный подход – формирование у учащихся осознанной самостоятельности в процессе приобретения предметных знаний и дальнейшей профессиональной деятельности.

Сектор образования в сфере высшего, среднего и дополнительного профессионального образования выпускает намного меньше выпускников по ИТ-специальностям, чем нужно быстро растущему ИТ-рынку. Причем качество подготовки специалистов приводит к тому, что только 30% из выпускников соответствуют необходимым требованиям. По предварительным подсчетам, в 2011 году спрос на выпускников необходимого уровня подготовки в сфере ИТ превысил их реальное количество в 2 раза (по материалам исследований компании IBS). 70% выпускников вузов через 2 года увольняют из компаний за профнепригодность несмотря на то, что в каждого выпускника на рабочем месте обычно тратится 1 год на формирование его профкомпетенций.[5]

Для решения проблем и противоречий в подготовке ИТ-специалистов необходимо не только искать пути взаимодействия ИТ-бизнеса и образовательных учреждений, но и продолжить программу формирования у ИТ-специалистов понимания важности систематического образования, постоянного самосовершенствования. Важно проводить работу, которая уже много лет проводится на Западе – демонстрировать ИТ-специалистам важность и ценность авторизованного обучения, полезность сертификации. И в эту работу необходимо включиться не только обучающим компаниям, но и компаниям-производителям программного обеспечения, и всему российскому бизнесу, заинтересованному в подготовленных качественных ИТ-кадрах.

Литература

1. Особенности обучения ИТ-специалистов: виртуальный круглый стол с экспертами. <http://itpractice.ru/itpractice/staff/948-itteaching.html> (дата обращения: 25.05.12).

2. Московский государственный университет имени М.В. Ломоносова. <http://www.msu.ru/info/struct/dep/vmc.html> (дата обращения: 25.05.12).
3. Инновационное образование в университете «Дубна». <http://grid2010.jinr.ru/files/pdf/che.pdf> (дата обращения: 26.05.12).
4. Система подготовки ИТ-специалистов нового качества на основе инновационной модели частно-государственного партнерства в вузах.
5. Доклад. М.И. Нежурина, А.Ю. Силантьев . Академия IBS (дата обращения: 27.05.12).
6. Рабочий альянс ИТ-компаний и вузов. <http://www.crn.ru/numbers/registration/detail.php?ID=19360> (дата обращения: 27.05.12).
7. Cisco развивает российское ИТ-образование. <http://www.cprice.ru/articles/detail.php?ID=458144> (дата обращения: 27.05.12).

346428, г. Новочеркасск, ул. Просвещения, 132, т.255-4-33 e-mail: rusalkaju@mail.ru

УДК 004

ОБЗОР ВОЗМОЖНОСТЕЙ ОБЛАЧНЫХ ТЕХНОЛОГИЙ

Горбатова Ю.А.

Южно-Российский государственный технический университет (НПИ), г. Новочеркасск

Рассмотрены возможности облачных технологий различных комплектаций и производителей, а также дальнейшие перспективы их развития.

Сегодня концепция облачных технологий и виртуализации стала очень популярной в сфере информационных технологий. "В сфере сетевых технологий это словосочетание является символом сегодняшнего времени", – говорит ведущий аналитик исследовательской и консалтинговой компании Gartner, специализирующейся на рынках информационных технологий, Бен Принг, и с ним соглашаются многие его коллеги.[1] Но при этом проблема состоит в том, что каждый трактует термин «облачная обработка данных» по-своему, хотя это лишь особый способ предоставления ИТ-ресурсов в виде сервиса, а не новая технология, как они вызвали революцию в методах предоставления информации и услуг. По идее, за счет улучшенной виртуализации машин, меньшего времени на администрирование и снижения затрат на инфраструктуру, снижаются расходы организаций, использующих облачные вычисления.

Облачные (рассеянные) вычисления – технология обработки данных, в которой компьютерные ресурсы и мощности предоставляются пользователю как Интернет-сервис. В отличие от традиционного подхода, когда аппаратные средства и программное обеспечение находятся на компьютере конечного пользователя, в облачных вычислениях программное обеспечение находится на удаленных серверах и предоставляется пользо-

вателю по необходимости, а аппаратные ресурсы выделяются по необходимости. Пользователь имеет доступ к собственным данным, но не может управлять и не должен заботиться об инфраструктуре, операционной системе и программном обеспечении, с которым он работает. [2]

В общем случае, облачная система и ее подписчики применяют модель клиент-сервер, где подписчики отправляют по сети запросы серверным компьютерам, которые в ответ на полученные сообщения выполняют соответствующую работу. При этом вычислительными ресурсами являются серверы, базы данных, файловые хранилища, которые расположены удаленно и объединены в общий пул мощностей. Очень часто такой подход изображают в виде облака из большого количества серверов.

Часто под словом "облако" понимают весь Интернет: то есть, все, что находится за пределами брандмауэра компьютера. Облачная система может функционировать в одной из четырех моделей:

Частное облако – инфраструктура, используемая одной организацией, которая может включать несколько потребителей. Например, подразделения одной организации, клиенты и подрядчики данной организации. Такое облако может находиться в собственности, управлении и эксплуатации как самой организации, так и третьей стороны, и располагаться как на территории предприятия, так и за его пределами.

Публичное облако – инфраструктура, свободно используемая широкой публикой, которая может быть не связана общими интересами, но принадлежащая к одной области деятельности. Публичное облако может находиться в собственности и эксплуатации коммерческих, научных и правительственных организаций, продающей соответствующие облачные услуги и предоставляющей облачные сервисы.

Общественное облако – облачная инфраструктура используется совместно несколькими организациями и поддерживает ограниченное сообщество, разделяющими общие принципы (например, миссию, требования к безопасности, политики, требования к соответствию регламентам и руководящим документам). Такая облачная инфраструктура может управляться самими организациями или третьей стороной и может существовать как на стороне потребителя так и у внешнего провайдера.[3]

Гибридное облако – это композиция из двух или более различных облачных инфраструктур (частных, публичных или общественных), которые остаются уникальными, но связаны между собой стандартизованными или частными технологиями, которые позволяют переносить данные или приложения между компонентами (например, для балансировки нагрузки между облаками).

Для того, чтобы автоматизировать предоставление ИТ-сервисов из облака нужна их стандартизация в виде сервисной модели облачной среды. Под этим понимается наличие формализованного описания параметров сервиса, который потребитель может получать из облака. В настоящее время различают несколько основных типов сервисных моделей облаков: IaaS, PaaS, SaaS.[4]

Программное обеспечение как услуга (SaaS). Данная система подразумевает предоставление потребителю в использование приложений провайдера, которое работает в облачной инфраструктуре. Приложения доступны из различных клиентских устройств или через интерфейсы тонких клиентов, такие как веб-браузер (например, веб-почта) или интерфейсы программ.

Платформа как услуга (PaaS). Облако предоставляет потребителю возможность для развертывания в облачной инфраструктуре потребительских приложений, которые реализуются с помощью языков программирования, библиотек, служб и средств, поддерживаемых провайдером услуг. Потребитель при этом не управляет базовой инфраструктурой облака, в том числе сетями, серверами, операционными системами и системами хранения данных, но имеет контроль над развернутыми приложениями и некоторыми параметрами конфигурации среды хостинга.

Инфраструктура как услуга (IaaS). Предоставляет потребителю возможность использования систем обработки, хранения, сетей и других фундаментальных вычислительных ресурсов для развертывания и запуска произвольного программного обеспечения, которое может включать в себя операционные системы и приложения. Потребитель при этом не управляет базовой инфраструктурой облака, но имеет контроль над операционными системами, системами хранения, развернутыми приложениями и ограниченный контроль выбора сетевых компонентов. Для того, чтобы развернуть приложение в IaaS облаке потребителю необходимо настроить все компоненты программного стека, начиная с операционной системы.[4]

В числе наиболее популярных «облачных» сервисов можно выделить представителей: Google Docs от поисковой системы Google; Office Web Apps от корпорации Microsoft; iCloud от корпорации Apple; Dropbox от одноименной компании; Amazon Cloud Drive от компании Amazon; Pixlr от компании Autodesk. Так же по мнению компании Parallels, специализирующейся на разработке программного обеспечения виртуализации и автоматизации, в ближайшие 5-10 лет большая часть ИТ переместится в облака пяти различных типов, где облака будут предоставлять различные платформенные услуги, – Google, Microsoft и другие крупные ИТ игроки: IBM, Apple,

HP и Amazon. Будут облака услуг, где планируется возникновение тысяч облачных провайдеров, предлагающих широкий спектр услуг. Например, веб-хостинг и хостинг приложений, вертикально интегрированные структуры независимых производителей программного обеспечения, телекоммуникационные услуги. И, наконец, будут внутренние облака крупных компаний, которые будут предоставлять услуги для внутреннего использования.

При использовании облачных вычислений потребители информационных технологий могут существенно снизить расходы на построение центров обработки данных, закупку серверного и сетевого оборудования, аппаратных и программных решений по обеспечению непрерывности и работоспособности.[5]

Кроме того, длительное время построения и ввода в эксплуатацию крупных объектов инфраструктуры информационных технологий и высокая их начальная стоимость ограничивают способность потребителей гибко реагировать на требования рынка, тогда как облачные технологии обеспечивают возможность практически мгновенно реагировать на увеличение спроса на вычислительные мощности.

При использовании облачных вычислений затраты потребителя смещаются в сторону операционных – таким образом классифицируются расходы на оплату услуг облачных провайдеров. Но есть также и противоположное мнение о применении облачных технологий, где считается, что использовать облачные веб-приложения для своих вычислительных процессов не следует потому, что пользователь теряет над ними контроль. И это не лучше, чем использовать любую проприетарную программу или чужой веб-сервер, так как в этом случае пользователь становится беззащитным.

В наше революционное время облачные вычисления могут предоставить организациям средства и методы, необходимые для обеспечения финансовой стабильности и высокого уровня обслуживания. Естественно, для достижения оптимальной защищенности облачных вычислений и выработки общих эксплуатационных стандартов необходима глобальная кооперация.

Литература

1. Что в действительности представляют облачные сервисы. http://www.hwp.ru/articles/CHto_v_deystvitelnosti_predstavlyayut_soboy_oblachnie_servisi/ (дата обращения: 23.05.12)
2. Заоблачные вычисления. <http://www.xakep.ru/magazine/xa/125/022/1.asp> (дата обращения: 23.05.12)
3. Референтная (эталонная) архитектура облачных вычислений. http://cloud.sorlik.ru/reference_architecture.html (дата обращения: 23.05.12)
4. Сервисные модели облачных сред. <http://www.smartnet.ru/node/119> (дата обращения: 24.05.12)
5. Облачные вычисления. <http://ru.wikipedia.org/wiki/> (дата обращения: 25.05.12)

6. Что такое облачные вычисления <http://www.parallels.com/ru/spp/understandingclouds/> (дата обращения: 25.05.12)
7. Основы облачных вычислений http://blog.i-oblako.ru/2012/03/blog-post_5366.html (дата обращения: 26.05.12)

346428, г. Новочеркасск, ул. Просвещения, 132, т.255-4-33, e-mail: rusalkaju@mail.ru

УДК 004.056

ОТДЕЛЬНЫЕ АСПЕКТЫ МЕТОДИКИ ПОСТРОЕНИЯ СИСТЕМЫ ЗАЩИТЫ ПЕРСОНАЛЬНЫХ ДАННЫХ

Горбатова Ю.А.

Южно-Российский государственный технический университет (НПИ), г. Новочеркасск

Рассмотрены отдельные аспекты методики построения системы защиты персональных данных на основе Федерального закона № 152 «О персональных данных» компаний Cisco, «Микротест», ActivIdentity.

Новые законы, регулирующие деятельность по обработке персональных данных, оказывают влияние как на рынок средств ИТ, так и на предлагаемые на нем технологические решения. Конечно назрела реальная необходимость обеспечивать адекватную защиту персональных данных. С каждым днем возрастает как ценность информации, так и ловкость способов, которыми ее можно незаконно получить. По материалам исследования компании InfoWatch в 2009 году среди всех зарегистрированных утечек информации персональные данные составили 89,8%. [1] На этом фоне государство принимает решение ужесточить ответственность за халатное отношение к хранению личных данных. 26.07.2006 Правительством РФ принят Федеральный закон № 152 «О персональных данных», согласно которому все организации, обрабатывающие сведения, относящиеся к персональным данным, не позднее 1 июля 2011 года должны привести свои информационные системы по обработке и защите персональных данных в соответствие с его требованиями [2]. По закону каждой информационной системе, где хранятся и обрабатываются персональные данные, необходимо присвоить класс, в соответствии с ним оказывать защиту этих сведений. Класс информационных систем определяется на основе модели угроз безопасности данных в соответствии с нормативно-методическими документами регуляторов.

Теперь, по новому российскому законодательству, компаниям, хранящим персональные данные своих клиентов, в случае их утери или защиты не должным образом, грозят не только дисциплинарные взыскания и штрафы, но и уголовная ответственность.

В связи с требованиями, которые предъявляет государство для организаций, возникает необходимость в построении такой подсистемы информационной безопасности, которая соответствовала бы требованиям руководящего документа Гостехкомиссии России: "Автоматизированные системы. Защита от несанкционированного доступа к информации. Классификация автоматизированных систем и требования по защите информации". Автоматизированная система в защищенном исполнении реализует информационную технологию выполнения установленных функций, таких как предупреждение о появлении угроз безопасности информации; обнаружение, нейтрализацию и локализацию воздействия угроз безопасности информации; управление доступом к защищаемой информации; восстановление системы защиты информации и защищаемой информации после воздействия угроз; регистрацию событий и попыток несанкционированного доступа к защищаемой информации и несанкционированного воздействия на нее; обеспечение контроля функционирования средств и системы защиты информации и немедленное реагирование на их выход из строя в соответствии с требованиями стандартов и нормативных документов по защите информации.

Защиту персональных данных можно обеспечить только в той информационной системе, где злоумышленник не может вмешаться в работу ее базовых элементов: сетевых устройств, операционных систем, приложений и СУБД. Базовыми инструментами для гарантии безопасности этого являются антивирусы, межсетевые экраны, системы предотвращения вторжений, сканеры уязвимостей. Средства защиты являются общими для всей сети и не связаны непосредственно с защитой персональных данных, однако в требованиях Федеральной службы по техническому и экспортному контролю их наличие специально оговаривается, поэтому эти базовые средства должны быть у каждого оператора персональных данных [5].

Перед тем как организации реализовать проект по защите персональных данных, появляется дополнительный вопрос: выполнять задачу собственными силами или привлекать стороннюю организацию. Компания может сама выполнить данные виды работ при наличии в штате специалистов по информационной безопасности, но проведение работ собственными силами может растянуться на длительный период времени, и из-за временных ограничений, установленных законом, возникают риски предъявления претензий со стороны регуляторов за неисполнение требований законодательства. При выборе сторонней организации необходимо учитывать основные критерии, по которым производится отбор исполнителя. А именно такими критериями являются обладание опытом и знаниями веде-

ния проектов по обеспечению информационной безопасности как в целом, так и защите персональных данных; наличие лицензий и необходимых ресурсов для участия в проекте; наличие успешно завершенных проектов по защите персональных данных.

Больше всего нуждаются в системе защиты те компании, которые оказывают услуги частным лицам с заключением договоров. Спектр этих предприятий, весьма велик: банки, страховые компании, сотовые операторы, интернет-провайдеры, турагентства, риелторы, нотариусы, железнодорожные и авиаперевозчики, все медицинские учреждения, учреждения образования и другие. Финансовые организации оказываются в непростой ситуации, так как они должны максимально защитить систему от атак злоумышленников, с другой – предоставить своим клиентам наиболее простые методы аутентификации для удобства. Решение такой проблемы предлагает лидер в области защиты персональных данных компания ActivIdentity в виде использования платформы 4TRESS Appliance, где были объединены передовые технологии строгой аутентификации и фрод-мониторинг в режиме реального времени. С помощью 4TRESS Appliance банки получают возможность развертывания многоуровневой системы строгой аутентификации и многократного повышения безопасности более удобным и доступным способом как в сети, так и в «облаке» [3].

Под особым вниманием со стороны регулирующих органов оказываются государственные организации или организации со значительной долей государственного участия; организации нефтегазовой отрасли; организации, обрабатывающие данные, касающиеся расовой, национальной принадлежности, политических взглядов, религиозных убеждений, состояния здоровья; организации, обрабатывающие данные более чем 100 000 субъектов персональных данных [4]. Российская ИТ-компания, которая специализируется на предоставлении услуг ИТ-консалтинга, внедрении бизнес-приложений, системной интеграции «Микротест» предлагает поэтапный подход к решению задач по защите, который состоит из следующих шагов: предпроектный анализ систем обработки персональных данных; подготовка внутренних документов в области защиты персональных данных; создание информационных систем защиты персональных данных; организация работы оператора по обработке персональных данных; аттестация информационной системы персональных данных в соответствии с указанным классом [4].

Решения Cisco в области защиты объединены в стратегию самозащищающейся сети Cisco Self-Defending Network, которая является частью концепции «Сети без границ» (Borderless Network). Ее идея достаточно

проста – это предоставление своевременного доступа законным пользователям с одновременным выявлением и предотвращением вторжений и нарушителей безопасности на всех уровнях информационной системы. Вместо того чтобы поддаваться атакам, инфраструктура отражать их и параллельно сохраняет доступность, надежность и отказоустойчивость.

Вступление в силу Федерального закона от 27.07.2006 г. № 152-ФЗ «О персональных данных» сформировало разные точки зрения относительно защиты персональных данных в России. С одной стороны это значительный шаг на пути к формированию нормативной правовой базы в сфере защиты информации, это хорошая возможность для России стать равноправным участником многих мировых экономических и политических процессов. Но с другой стороны реализация закона в полном объеме невозможна, так как необходимо провести большой объем работ для устранения неточностей некоторых его моментов.

Литература

1. Закон о персональных данных касается всех. <http://www.cnews.ru/reviews/index.shtml?2010/08/09/404375>
2. Защита персональных данных. Почему защита персональных данных сегодня особенно актуальна? <http://services.softline.ru/pdp>
3. Многоуровневая защита для банков от Acti vIdentity <http://www.rnbo.ru/press-center/news839.php>
4. Защита персональных данных. Федеральный закон № 152-ФЗ http://www.microtest.ru/hardware/information_security/2687/
5. Защита персональных данных: проблемы и решения <http://www.osp.ru/os/2009/06/10050193/>

УДК 519.85:004.942

ОПТИМИЗАЦИЯ РАЗМЕЩЕНИЯ ФРАГМЕНТОВ БАЗЫ ДАННЫХ ПО УЗЛАМ СЕТИ С ОБЛАЧНОЙ АРХИТЕКТУРОЙ ПО КРИТЕРИЮ МИНИМУМА СТОИМОСТИ ХРАНЕНИЯ

Горобец В.В.

Южно-Российский государственный технический университет (НПИ), г. Новочеркасск

Рассматривается задача выбора оптимального распределения фрагментов базы данных по узлам сети с облачной архитектурой по критерию минимума общей стоимости хранения и максимума суммарной ценности реплик фрагментов. При этом учитываются ограничения на объем доступного дискового пространства.

В последнее время понятие облачных вычислений (cloud computing) пользуется чрезвычайно большой популярностью. Компания Gartner ста-

вит их в верхние строчки списка наиболее прорывных технологий ближайших лет [1].

Облачные вычисления имеют ряд преимуществ для развертывания приложений, оперирующих большими объемами данными. Одним из главных достоинств является практически неограниченная производительность, достигаемая путем увеличения числа узлов при возрастании рабочей нагрузки. Другая важная перспектива состоит в снижении расходов на основе использования бизнес-модели «платы по мере использования» (pay-as-you-go) [2]. За счет устранения или упрощения процессов планирования, приобретения и подготовки к использованию аппаратных средств и перехода к использованию такой модели капитальные вложения можно превратить в текущие расходы. Кроме того, облачная инфраструктура может обеспечить более высокий (близкий к 100%) коэффициент загрузки аппаратных ресурсов по сравнению с GRID-архитектурой [3]. Поэтому облачные вычисления часто считают важной технологией для обеспечения экологически безопасного использования вычислительных средств (green computing). Кроме того, ее использование сокращает эксплуатационные расходы и трудозатраты за счет автоматизации таких ИТ-задач, как обновление системы безопасности и переключение на резервные мощности при отказе оборудования (fail-over). Наконец, модель облачных вычислений обеспечивают гибкость при использовании программных и аппаратных средств и при управлении ими, что позволяет сократить время выхода на рынок развивающимся компаниям и уменьшить расходы.

В связи с тем, что в последнее время наблюдается тенденция, связанная с развертыванием OLTP-систем на базе облачной архитектуры, предполагающей использование бизнес-модели «плата по мере использования», возникает задача рационального вложения средств в аренду аппаратных ресурсов. Существует большое количество работ, посвященных решению задачи оптимизации размещения информационных массивов по узлам сети. В частности, в монографии [4] как одна из важных упоминается проблема распределения данных по накопителям памяти на серверах компьютерной сети. В работе [5] решается задача оптимального размещения данных по файловым серверам, минимизирующего среднее время доступа для случая, когда стоимость размещения и хранения единицы данных является линейно возрастающей функцией от их объема. Однако появление облачной архитектуры требует пересмотра подходов к решению поставленной задачи с учетом особенностей описанной технологии.

Поэтому в рамках данной работы рассматривается задача выбора оптимального размещения фрагментов базы данных по узлам сети с облачной архитектурой, минимизирующего общую стоимость хранения. При этом учитываются ограничения на объем доступного дискового пространства и на обеспечение максимального коэффициента загрузки аппаратных ресурсов.

Пусть имеется сеть (рис. 1) с произвольной топологией, соединяющая узлы, которые состоят из ЭВМ и сетевого адаптера, и локальная вычислительная сеть (ЛВС) с регулярной структурой, все ЭВМ которой имеют доступ посредством канала связи в первую сеть. Архитектура сети с произвольной топологией описывает облачную структуру.

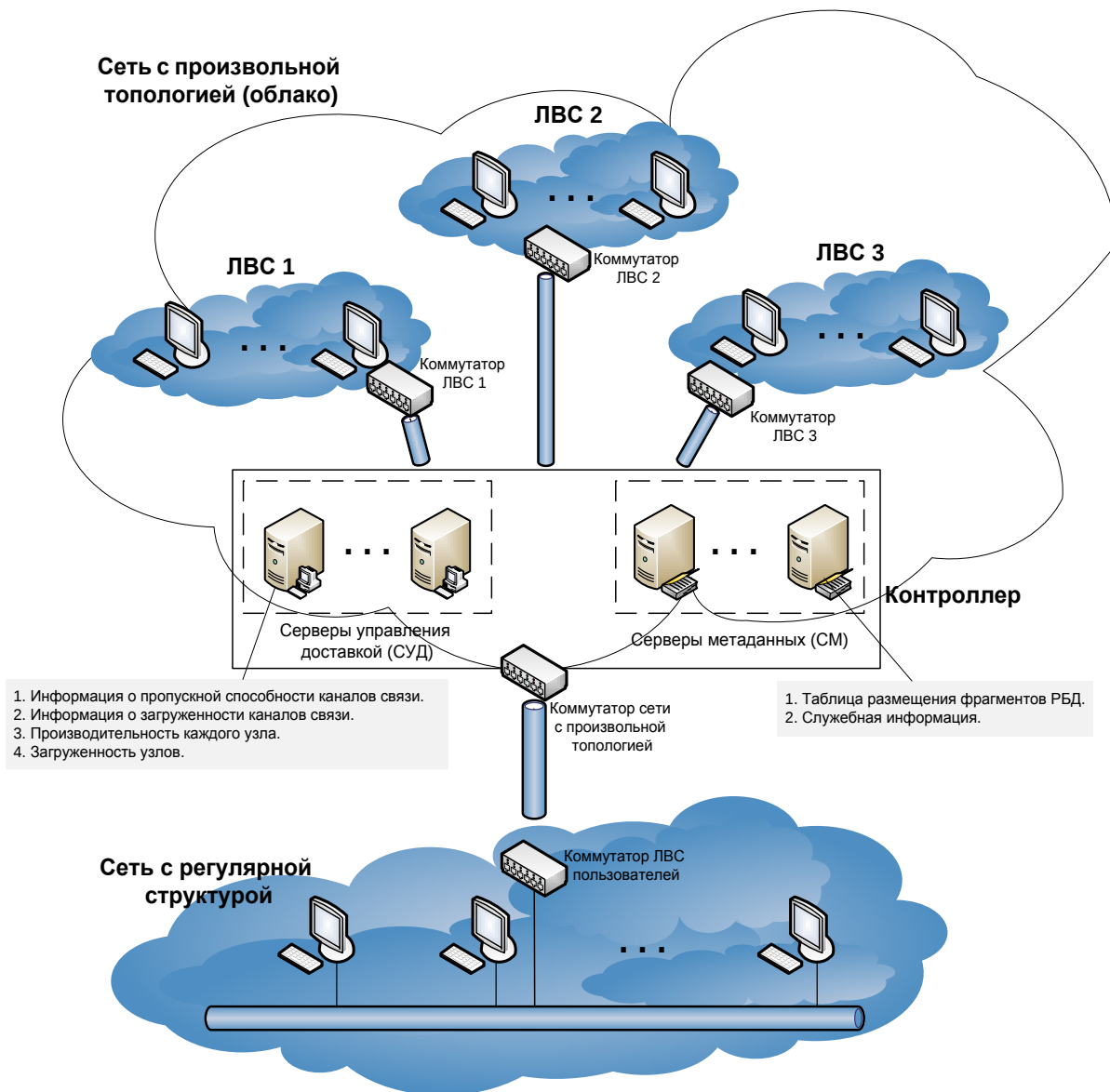


Рис. 1. Сетевая структура

Кроме того, имеется база данных, распределенная по узлам сети с произвольной топологией. Необходимо таким образом распределить фрагменты базы данных по узлам, входящим в облако, чтобы суммарная стоимость аренды аппаратных ресурсов была минимальной и выполнялись ограничения, связанные с объемом доступной памяти каждого узла. В рамках данной модели в качестве арендуемых аппаратных ресурсов будем считать дисковое пространство. Так как в рассматриваемой бизнес-модели «плата по мере использования» стоимость хранения информационных массивов является линейно возрастающей функцией от их объема, то с учетом необходимости повышения доступности и надежности данных задача сводится к определению оптимального количества копий (реплик) фрагментов базы данных и получению варианта их размещения.

Пусть

n – число узлов сети с произвольной структурой;

m – число независимых фрагментов РБД;

K_j – j -ый узел сети, $j = \overline{1, n}$;

F_i – i -ый фрагмент РБД, $i = \overline{1, m}$;

L_i – объем i -го фрагмента;

b_j – объем памяти узла K_j , предназначенной для размещения фрагментов;

r – стоимость хранения единицы информации в облаке.

Введем матрицу $X = \|x_{ij}\|$ прикрепления фрагментов РБД к конкретным узлам сети с облачной архитектурой, $i = \overline{1, m}$, $j = \overline{1, n}$, $z = \overline{1, n}$, где $x_{ij} \geq 0$ – количество реплик фрагмента F_i , находящихся в узле K_j .

Тогда для каждого K_j -го узла суммарный объем b_j дискового пространства, необходимого для размещения в нем фрагментов, не должен превышать объем доступной памяти этого узла:

$$\sum_{i=1}^m L_i x_{ij} \leq b_j, \quad j = \overline{1, n}. \quad (1)$$

Стоимость хранения фрагмента F_i вычисляется по формуле:

$$C_i = L_i r, \quad i = \overline{1, m}.$$

При этом общая стоимость C размещения всех фрагментов определяется выражением:

$$C = \sum_{i=1}^m \sum_{j=1}^n C_i x_{ij}. \quad (2)$$

Также необходимо обеспечить доступность и надежность данных путем использования реплик фрагментов. Учет данного требования дости-

гается за счет введения коэффициента ценности KC_i фрагмента, который, в соответствии с [6], предлагается рассчитывать, исходя из:

- популярности фрагмента R_i , то есть количества обращений к фрагменту за определенный период времени;
- степени важности фрагмента I_i ;
- размера фрагмента T_i .

Для учета значимости этих параметров важность каждого из них оценивается в баллах (соответственно w_{pop} , w_{imp} , w_{size} (экспертная оценка)).

Степень важности фрагмента I_i предлагается выражать в общем случае как некоторую функцию:

$$I_i = f(P_{lost F_i}), i = \overline{1, m},$$

где $P_{lost F_i}$ – вероятность потери i -го фрагмента, задаваемая в качестве исходных данных.

На основе вектора весовых коэффициентов и значений параметров вычисляется коэффициент ценности фрагмента F_i :

$$KC_i = w_{pop} \frac{R_i}{R_{max}} + w_{imp} \frac{I_i}{I_{max}} + w_{size} \frac{T_i}{T_{max}}, i = \overline{1, m},$$

где

R_{max} , I_{max} , T_{max} – максимальные значения параметров;

w_{pop} , w_{imp} , w_{size} – весовые коэффициенты;

m – число фрагментов.

Тогда учет требования обеспечить доступность и надежность данных можно выразить через суммарную ценность KC всех фрагментов РБД:

$$KC = \sum_{i=1}^m \sum_{j=1}^n KC_i x_{ij}. \quad (3)$$

Таким образом, рассматриваемая задача оптимального распределения фрагментов по узлам вычислительной сети состоит в том, чтобы определить значения переменных x_{ij} , где $x_{ij} \geq 0$ ($i = \overline{1, m}$; $j = \overline{1, n}$), которые удовлетворяют условию (1) и дают минимум линейной функции (2) при максимуме линейной функции (3), то есть:

$$\begin{aligned} C(X) &= \sum_{i=1}^m \sum_{j=1}^n C_i x_{ij} \rightarrow \min, \\ KC(X) &= \sum_{i=1}^m \sum_{j=1}^n KC_i x_{ij} \rightarrow \max, \\ \sum_{i=1}^m L_i x_{ij} &\leq b_j, x_{ij} \geq 0, i = \overline{1, m}, j = \overline{1, n}. \end{aligned} \quad (4)$$

Полученная математическая модель является задачей целочисленного линейного программирования, которая на концептуальном уровне сводится к задаче о ранце. Для ее решения разработано большое число точных и приближенных методов [7]. Одним из наиболее распространенных точных методов является метод ветвей и границ (МВГ), который состоит в декомпозиции множества допустимых решений на подмножества (подзадачи) с отсевом подмножеств, заведомо не содержащих оптимума. Существуют реализации данного подхода, предложенные Горовицем, Сахни [7] и Писингером [8]. Задача о ранце также может быть решена методами дискретного программирования, симплекс-методом, методом Гомори [9].

Исходя из представленной модели размещения, можно сказать, что максимальный коэффициент загрузки аппаратных ресурсов достигается путем варьирования числа n задействованных узлов сети облачной структуры. Это означает, что поставленная задача в общем случае решается для разных n и по полученным результатам выбирается такой вариант распределения, при котором этот коэффициент наибольший.

В заключение стоит отметить, что при увеличении количества реплик фрагментов растет доступность, надежность и скорость доступа к ним. С другой стороны, увеличение числа копий приводит к росту затрат памяти, а соответственно, и стоимости аренды ресурсов. Так возникает задача управления репликами, решение которой снижает необоснованные затраты. В рамках данной статьи была поставлена задача выбора оптимального распределения фрагментов базы данных по узлам сети с облачной архитектурой по критерию минимума общей стоимости размещения (с точки зрения аренды дискового пространства) и максимума суммарной ценности реплик фрагментов с учетом ограничения на объем доступной памяти.

Литература

1. Gartner. Gartner Top Ten Disruptive Technologies for 2008 to 2012. Emerging Trends and Technologies Roadshow, 2008.
2. Коссман Д., Краска Т., Лоузинг С. Анализ альтернативных архитектур управления транзакциями в «облачной» среде. URL: http://citforum.ru/database/articles/kossmann_sigmod_2010/2.shtml (дата обращения: 29.11.2010).
3. Семенов Ю.А. Сети GRID. URL: <http://book.itep.ru/4/7/grid.htm> (дата обращения: 27.12.2010).
4. Столлингс В. Компьютерные системы передачи данных / В. Столлингс. – М.; СПб.; К.: Вильямс, 2002. – 920 с.
5. Гуревич И.М. Об оптимальном размещении данных по файловым серверам вычислительной системы / И.М. Гуревич, В.В. Кирюхин, А.В. Герус // Материалы междунар. науч.-практ. конф. «ИНФОТЕХ–2007» 10–16 сентября 2007 г. – Севастополь, 2007. – Ч. 1. – С. 13–16.

6. Лукьянов Н.М. Разработка архитектуры и методов организации слабосвязанных архивных систем для автоматизации проектирования: дис. канд. техн. наук : 05.13.12. – СПб.: СПбГУ ИТМО, 2011. – 85 с.
7. Martello S., Totli P. Knapsack Problems: Algorithms and Computer Implementations. John Wiley & Sons. – 1990.
8. Kellerer H., Pferschy U., Pisinger D. Knapsack Problems. Berlin. Germany: Springer. – 2004.
9. Сигал И.Х., Иванова А.П. Введение в прикладное дискретное программирование: модели и вычислительные алгоритмы. М: ФИЗМАТЛИТ. – 2007.

346428, г. Новочеркасск, ул. Просвещения, 132, т.255-4-33, e-mail: vtl88@mail.ru

УДК 519.85:004.942

ОПТИМИЗАЦИЯ РАЗМЕЩЕНИЯ ФРАГМЕНТОВ РБД В УЗЛАХ СЕТИ С ПРОИЗВОЛЬНОЙ ТОПОЛОГИЕЙ В РАМКАХ ОБЛАЧНОЙ СТРУКТУРЫ ПО КРИТЕРИЮ МИНИМУМА СТОИМОСТИ ТРАФИКА

Горобец В.В.

Южно-Российский государственный технический университет (НПИ), г. Новочеркасск

Рассматривается задача оптимального размещения фрагментов распределенных баз данных в узлах произвольной сети системы, построенной по принципу облачной структуры. В качестве критерия оптимальности принимается общая стоимость трафика, порожденного функционированием вычислительной системы в течение единицы времени. Предлагаются модели и схемы фрагментации.

Технологии баз данных (БД) в настоящее время используются практически во всех организациях. Следствием возрастающей сложности систем обработки и представления информации является увеличение потребностей в управлении все большими объемами данных. При этом задачи информационной интеграции БД и проектирования географически распределенных БД (РБД) являются наиболее актуальными для разработчиков программного обеспечения в течение почти трех десятилетий [1].

Базы данных занимают центральное место в автоматизированных информационно-управляющих системах (АИУС). От эффективности и качества их построения во многом зависит эффективность разрабатываемых распределенных информационных систем [2]. Поэтому разработка математических моделей для систем автоматизированного проектирования (САПР) БД является важной и актуальной задачей.

В последнее время наблюдается тенденция, связанная с развертыванием OLTP-систем на базе облачной архитектуры, предполагающей ис-

пользование бизнес-модели «плата по мере использования». В связи с этим возникает задача рационального вложения средств в аренду аппаратных ресурсов. Так, в статье [3] предлагается математическая модель оптимального размещения фрагментов РБД, где качестве критерия оптимальности выступает средний объем пересылаемых по линиям связи данных при выполнении запроса. в рамках данной работы также рассматривается метод построения модели оптимального размещения фрагментов РБД в вычислительной сети с произвольной топологией в рамках облачной структуры. Однако в качестве критерия оптимальности возьмем общую стоимость трафика, порожденного функционированием вычислительной системы в течение единицы времени.

Пусть имеется сеть (рис. 1) с произвольной топологией, соединяющая узлы, которые состоят из ЭВМ, и локальная вычислительная сеть (ЛВС) с регулярной структурой, все ЭВМ которой имеют доступ посредством канала связи в первую сеть. Архитектура сети с произвольной топологией отражает облачную структуру.

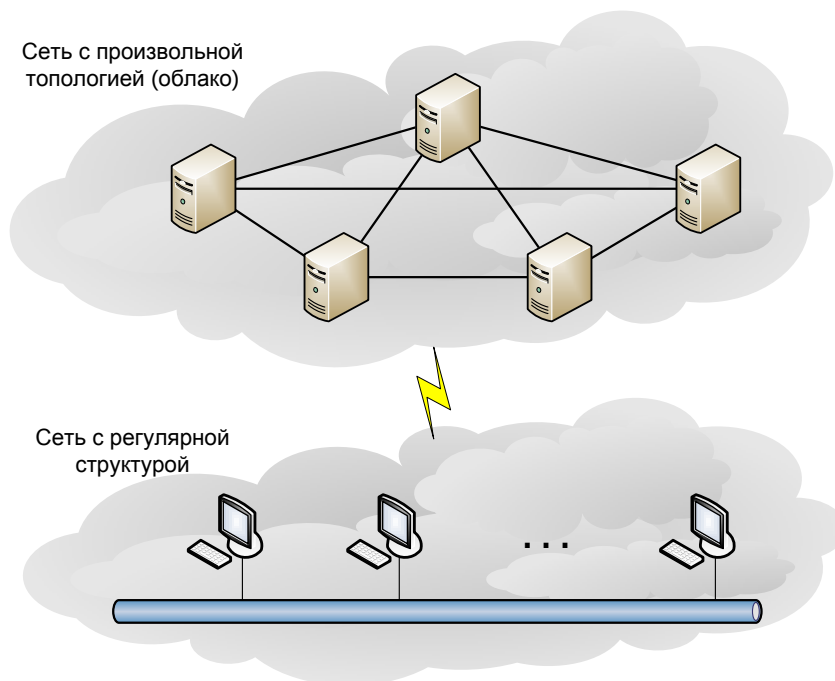


Рис. 2. Сетевая структура

Предположим, что запрос, поступающий на любой узел сети с произвольной топологией, предполагает доступ к определенному фрагменту РБД, и все запросы к одному и тому же фрагменту могут иметь различную длину и требуют для ответа различный объем данных. Запрос, инициированный на терминале в сети регулярной структуры, поступает во входную

очередь узла сети с произвольной топологией. Процессор ЭВМ обрабатывает запросы в порядке их поступления. Если нужный фрагмент содержится в локальной базе данных узла, на который поступил запрос, то запрос обрабатывается и результат выводится на инициировавший данный запрос терминал. Если нужный фрагмент содержится в другом узле, то запрос пересылается в этот узел, там обрабатывается и результат пересылается в первоначальный узел. Порядок обслуживания запросов в узлах не влияет на объем пересылаемых данных по каналам связи. Поэтому он может быть принят таким же, как и в случае модели для звездообразной топологии [4].

Из описанной схемы обработки запросов в системе следует, что в процессе обслуживания в течение каждой единицы времени по каналам связи пересылается некоторый объем данных. Общее значение объема данных, пересылаемых по каналам связи в рамках облачной структуры, зависит от распределения фрагментов по локальным базам данных сети с произвольной топологией. Чем меньше средний объем пересылаемых данных по каналам сети за единицу времени, тем ниже общая стоимость трафика. Он будет минимален, если на каждом узле будет находиться полный набор фрагментов РБД. Но для больших баз данных это практически недостижимо, так как объемы памяти узлов ограничены.

Запишем целевую функцию задачи оптимального распределения фрагментов по узлам вычислительной сети в случае, когда запросы из фиксированного узла к одному и тому же фрагменту могут быть различного типа.

Пусть

n – число узлов сети с произвольной структурой;

m – число независимых фрагментов РБД;

K_j – j -ый узел сети, $j = \overline{1, n}$;

F_i – i -ый фрагмент РБД, $i = \overline{1, m}$;

L_i – объем i -го фрагмента;

b_j – объем памяти узла K_j , предназначенной для размещения фрагментов;

λ_{ij} – интенсивность запросов к фрагменту F_i , инициированных в узле K_j ;

α_{ij} – объем запроса к фрагменту F_i , инициированного на терминале узла K_j ;

β_{ij} – объем запрашиваемых данных при выполнении запроса к фрагменту F_i , поступившего на терминал узла K_j ;

r_{ij} – стоимость передачи единицы информации из узла K_i в узел K_j , $i = \overline{1, n}$, $j = \overline{1, n}$ ($r_{ii} = 0$);

s – число классов запросов (например, чтение, добавление, обновление, удаление записей БД);

λ_{ij}^k – интенсивность запросов k -го типа к фрагменту F_i , инициированных в узле K_j ;

α_{ij}^k – объем запроса k -го типа к фрагменту F_i , инициированного на терминале узла K_j ;

β_{ij}^k – объем запрашиваемых данных при выполнении запроса k -го типа к фрагменту F_i , поступившего на терминал узла K_j .

Предлагаемая концепция фрагментации объединяет подходы горизонтального и вертикального разбиения и состоит в следующем. Фрагменты могут включать в себя не только построчные или построчные данные, но и представлять из себя произвольную комбинацию ячеек таблиц. Кроме того, один фрагмент может содержать информацию, хранящуюся в разных таблицах. На рис. 2 приведены примеры, поясняющие идею предложенной концепции фрагментации.

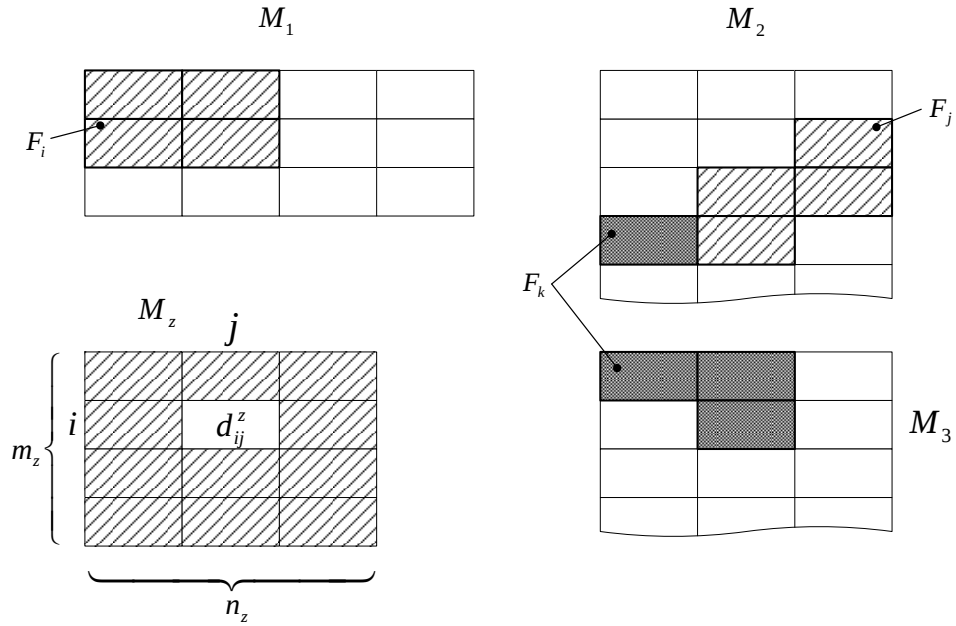


Рис. 3. Примеры схем фрагментации

Все сообщения, поступающие во входные очереди узлов, разделим на два типа: тип 1 – сообщения, составляющие запросы, для обработки которых необходимые фрагменты не содержатся в базе данных узла, и ответы на эти запросы; тип 2 – сообщения, составляющие запросы, для обслуживания которых нужные фрагменты содержатся в базе данных соответствующего узла. При этом будем считать, что запрос типа 1, прибывший для своего обслуживания в отдаленный узел, превращается в запрос типа 2.

Тогда объем пересылаемых данных при выполнении запроса k -го класса к фрагменту F_i , инициированного в узле K_j , равен $(\alpha_{ij}^k + \beta_{ij}^k)(1 - x_{ij})$. При этом $x_{ij} (i = \overline{1, m}; j = \overline{1, n})$ – величины, определяемые по формуле:

$$x_{ij} = \begin{cases} 1, & \text{если фрагмент } F_i \text{ находится в узле } K_j; \\ 0, & \text{в противном случае.} \end{cases}$$

Поскольку интенсивность λ_{ij}^k порождает объем данных $\lambda_{ij}^k (\alpha_{ij}^k + \beta_{ij}^k)(1 - x_{ij})$, нуждающихся в пересылке, то стоимость их пересылки равна:

$$C_{ij} = \lambda_{ij} (\alpha_{ij}^k + \beta_{ij}^k) \cdot (1 - x_{ij}) \cdot r_{ij},$$

а общая стоимость данных, которые следует переслать по линиям связи между узлами вследствие функционирования системы в течение единицы времени, определяется формулой:

$$C = \sum_{i=1}^m \sum_{j=1}^n \sum_{k=1}^s \lambda_{ij} (\alpha_{ij}^k + \beta_{ij}^k) \cdot (1 - x_{ij}) \cdot r_{ij}. \quad (1)$$

Поскольку каждый фрагмент $F_i (i = \overline{1, m})$ должен находиться в одном из узлов вычислительной сети, то

$$\sum_{j=1}^n x_{ij} = 1, \quad i = \overline{1, m} \quad (2)$$

Кроме того, объем локальной базы данных каждого узла $k_j (j = \overline{1, n})$ не должен превышать объем памяти этого узла, предназначенный для размещения фрагментов. Поэтому

$$\sum_{i=1}^m L_i x_{ij} \leq b_j, \quad j = \overline{1, n} \quad (3)$$

Таким образом, задача оптимального распределения фрагментов по узлам вычислительной сети по критерию минимизации общей стоимости трафика, порожденного функционированием вычислительной системы в течение единицы времени, состоит в том, чтобы определить значения переменных x_{ij} , где $x_{ij} = \{0; 1\}$ ($i = \overline{1, m}; j = \overline{1, n}$), которые удовлетворяют условиям (2)-(3) и дают минимум линейной функции (1). Полученная постановка математической модели является задачей целочисленного линейного программирования с булевыми переменными.

Если рассмотреть описанную задачу оптимизации более подробно, то необходимо учесть состав и структуру фрагментов. Для этого дополнительно введем следующие исходные данные:

p – число таблиц РБД;

M_z – z -ая таблица РБД, $z = \overline{1, p}$;

m_z – число строк таблицы M_z ;

n_z – число столбцов таблицы M_z ;

d_{ij}^z – ячейка таблицы M_z в позиции строка i , столбец j ;

w_{ij}^z – объем памяти, занимаемый ячейкой d_{ij}^z .

Далее запишем ограничения, связанные с распределением ячеек по фрагментам:

$$y_{ijz}^{\sigma} = \begin{cases} 1, & \text{если ячейка } d_{ijz} \text{ принадлежит фрагменту } F_{\sigma}; \\ 0, & \text{в противном случае.} \end{cases}$$

Суммарный объем всех фрагментов должен быть равен суммарному объему всех таблиц РБД, т.е.:

$$\sum_{i=1}^m L_i = \sum_{z=1}^p \sum_{i=1}^{m_z} \sum_{j=1}^{n_z} w_{ij}^z \quad (4)$$

$$\text{При этом объем фрагмента } F_{\sigma} \text{ равен: } L_{\sigma} = \sum_{z=1}^p \sum_{i=1}^{m_z} \sum_{j=1}^{n_z} w_{ij}^z \cdot y_{ijz}^{\sigma}. \quad (5)$$

Тогда на более детальном уровне задача состоит в том, чтобы найти значения переменных x_{ij} , где $x_{ij} \in \{0,1\}$ ($i = \overline{1, m}$; $j = \overline{1, n}$), которые удовлетворяют условиям (2)-(5) и дают минимум линейной функции (1).

Данную задачу можно решить с помощью генетического алгоритма. Тогда, исходя из постановки задачи, сформировать структуру хромосомы можно следующим образом (табл. 1, табл. 2):

Таблица 1

Схема фрагментации и размещения фрагментов по узлам

Фрагменты \ Узлы	Таблица 1			...	Таблица p		
	Фрагмент 1	...	Фрагмент m		Фрагмент 1	...	Фрагмент m
Узел 1	x_{11}	...	x_{m1}	...	x_{11}	...	x_{m1}
Узел 2	x_{12}	...	x_{m2}	...	x_{12}	...	x_{m2}
...	...	...	...	...	...	...	...
Узел n	x_{1n}	...	x_{mn}	...	x_{1n}	...	x_{mn}

Таблица 2

Схема фрагментации и размещения ячеек по фрагментам

Фрагменты \ Ячейки	Ячейки		
	Ячейка 1	...	Ячейка τ
Фрагмент 1	y_{11}	...	$y_{\tau 1}$

Фрагмент 2	y_{12}	...	$y_{\tau 2}$
...	...	...	...
Фрагмент m	y_{1m}	...	$y_{\tau m}$

Так как рассматриваемая концептуальная модель не предполагает использование репликации фрагментов базы данных, то в любом столбце таблиц 1, 2 может быть только одна единица, т.е. должны выполняться следующие условия:

1. Один фрагмент может быть расположен только на одном узле;
2. Одна ячейка может принадлежать только одному фрагменту.

В работе [3] подобная задача решалась с применением генетического алгоритма, результаты работы которого показали, что в некоторых случаях оптимальные значения целевой функции достигаются при различных вариантах схем фрагментации (т.е. различные решения поставленной задачи дают приблизительно одинаковые значения целевой функции). Кроме того, при увеличении числа узлов, фрагментов и типов запросов в исследуемой модели повышается вероятность возникновения дубликатов фрагментов на различных узлах. Это подтверждает тот факт, что использование репликации благоприятным образом сказывается на скорости обслуживания запросов в моделируемой системе. Анализируя вид целевой функции и состав ограничений модели, приведенной в рамках данной работы, и сравнивая их с соответствующими выражениями, представленными в статье [3], по аналогии можно сказать, что наличие копий фрагментов снизит общую стоимость трафика, порожденного функционированием вычислительной системы в течение единицы времени. Однако для подтверждения данного предположения и выявления особенностей рассмотренной модели необходимо провести дополнительные исследования. С другой стороны, увеличение числа реплик приводит к росту затрат памяти, а соответственно, и стоимости аренды ресурсов. Поэтому возникает задача выбора оптимального распределения фрагментов базы данных по узлам сети с облачной архитектурой по критерию минимума общей стоимости передаваемого трафика, стоимости хранения данных в облаке (с точки зрения аренды дискового пространства) и максимума суммарной ценности реплик фрагментов с учетом ограничения на объем доступной памяти узлов.

Литература

1. Новосельский В.Б. Методы автоматизации проектирования распределенных баз данных : дис. канд. техн. наук : 05.13.12. – СПб: СПбГУ ИТМО, 2008. – 114 с.

2. Ahmad I., Karlapalem K., Kwok Y.K., et al. Evolutionary Algorithms for Allocating Data in Distributed Database Systems // Distributed and Parallel Databases. – 2002. – Vol. 11, № 1. – pp. 5–32.
3. Горобец В.В. Модель оптимального размещения фрагментов РБД в узлах сети с произвольной топологией в рамках облачной структуры // Теория, методы проектирования, программно-техническая платформа корпоративных информационных систем: сб. по материалам 11-й Междунар. науч.-практ. конф., г. Новочеркасск, май 2011 г. – Новочеркасск : ЮРГТУ (НПИ), 2011. – С. 21-31.
4. Модели оптимального размещения таблиц РБД в узлах сети со звездообразной топологией. http://ami.nstu.ru/~vms/lecture/lecture11/Opt_replace1.htm (дата обращения: 21.03.2011).

346428, г. Новочеркасск, ул. Просвещения, 132, т.255-4-33, e-mail: vtl88@mail.ru

УДК 004.942:006.015.7

МОДЕЛЬ НАДЕЖНОСТИ ФУНКЦИОНИРОВАНИЯ ТРАНЗАКЦИОННОЙ СИСТЕМЫ, ПОСТРОЕННОЙ НА БАЗЕ ОБЛАЧНОЙ АРХИТЕКТУРЫ

Горобец В.В.

Южно-Российский государственный технический университет (НПИ), г. Новочеркасск

Рассматривается модель транзакционной системы, построенной по принципу облачной структуры, с точки зрения надежности функционирования. Приводится обобщенный алгоритм построения дерева отказов для определения показателей надежности. Предлагается модификация модели, учитывающая наличие дублирования каналов связи, сетевого оборудования и фрагментов распределенной базы данных.

Разработка современных информационных систем, особенно транзакционных, включает в качестве одного из обязательных этапов проектирования анализ их надежности. Проблема усложняется тем, что коммутационные сети, к анализу которых в конечном итоге сводится данная задача, являются сложными структурами. Это затрудняет, а порой делает невозможным расчет их надежности строго аналитическими методами, как это имеет место, например, для параллельно-последовательных сетей. Единственным численным методом расчета надежности сложных сетей остается метод полного перебора, который, однако, даже с привлечением высокопроизводительных ЭВМ, не позволяет анализировать сети, содержащие более десятков-сотен компонент (каналы связи, коммутаторы, серверы, рабочие станции).

При использовании технологии облачных вычислений, предполагающей перенос вычислений и обработку данных в существенной степени с персональных компьютеров на серверы сети Интернет, задача обеспечения надежности функционирования всей системы выходит на первый план. Одним из способов повышения надежности таких сетей является простое дублирование составляющих их элементов. Однако вследствие ограниченности ресурсов такой путь в большинстве случаев нерационален. К настоящему времени аналитический аппарат синтеза оптимальных структур коммуникационных сетей с облачной структурой недостаточно развит вследствие исключительной сложности самой задачи. В инженерной практике при решении подобного рода задач часто прибегают к методу частичного перебора [1]. Так, например, при выборе оптимальной структуры сети связи в качестве частных вариантов могут анализироваться некоторые типовые схемы соединения узловых пунктов. Применяется так называемый радиальный принцип соединения узлов, принцип связи «каждого с каждым» или «каждого с ближайшим», иерархический принцип соединения и т.д.

Существует большое число работ, посвященных сетям ЭВМ, в том числе ЛВС, в которых предлагаются различные модели и методы количественного обоснования принимаемых решений по выбору рациональных вариантов построения и организации функционирования ЛВС. К авторам наиболее известных этих работ относятся Глушков В.М., Флинт Д., Якубайтис А.Я. В работах этих авторов [2-6] излагаются основные принципы построения сетей ЭВМ, организация сетей передачи данных и связи, их анализ и синтез.

В работе [7] исследуется влияние топологических характеристик сетей на их надежность, пропускную способность, стоимость и ряд других системных характеристик. В [8] рассматривается проблема оптимизации физической структуры информационно-вычислительных сетей. Методы и алгоритмы синтеза, оптимизации структуры и повышения надежности функционирования централизованных и распределенных сетей ЭВМ с единых методологических позиций рассмотрены в работе [9].

Для обеспечения транзакционной надежности OLTP-систем, базирующихся на классической трехзвенной архитектуре, предложено множество подходов, включая организационные методы разработки, различные технологии и технологические программные средства. Следует отметить работы как российских, так и зарубежных исследователей в данной обла-

сти: Васкевича Д., Голосова А.О., Фагина Р. (Fagin R.), Мосса Дж. (Moss J.E.B.) и др.

Если рассмотреть накопленный опыт при решении задач оптимизации и повышения устойчивости информационно-вычислительного процесса, сохранности информации в сетях ЭВМ, то необходимо отметить научный вклад таких ученых, как Хорошевский В.В. [10], Турута Е.Н. [11-12], Кондратьев О.К. [13], Мамиконов А.Г., Кульба В.В. [14]. Авторы этих работ, связанных с оптимизацией информационно-вычислительного процесса в системе вычислительных средств сети ЭВМ, рассматривают программное и информационное обеспечение на уровне функциональных модулей, построенных без учета свойств распределенных систем. При этом общим недостатком является то, что модульность, положенная в основу решения задач рациональной организации сбора и обработки информации, распределения и перераспределения программных модулей и информационных массивов по узлам сети, принимается как данность ввиду ряда трудностей, связанных с их решением.

В настоящее время наблюдается направление, связанное с развертыванием OLTP-систем на базе облачной архитектуры. Разнородность составляющих их элементов по различным качественным и количественным параметрам в значительной степени влияет как на характеристики, так и на эффективность функционирования всей информационной системы. Поэтому современные тенденции к расширению задач, решаемых транзакционными системами, усложнили выполнение растущих требований, предъявляемых к надежности и безопасности бизнес-приложений, в частности, биллинговых. Удовлетворение этих требований в условиях расширения спектра дефектов программных и аппаратных средств, вызывающих нарушение их работоспособности, может обеспечиваться за счет адекватного изменения архитектур и технологий, поддерживающих отказоустойчивость.

В связи с этим в рамках данной работы с точки зрения надежности функционирования рассматривается модель транзакционной системы, построенной по принципу облачной структуры. При этом предполагается, что в вычислительной сети с произвольной топологией (облаке) размещены фрагменты распределенной базы данных (РБД). В качестве показателя надежности функционирования информационной системы примем вероятность возникновения отказа в облаке.

Пусть имеется сеть (рис. 1) с произвольной топологией, соединяющая узлы, которые состоят из ЭВМ и сетевого адаптера, и локальная вы-

числительная сеть (ЛВС) с регулярной структурой, все ЭВМ которой имеют доступ посредством канала связи в первую сеть. Архитектура сети с произвольной топологией описывает облачную структуру. Сначала определим надежность той части системы, которая расположена в облаке, в случае, когда отсутствует дублирование каналов связи, сетевого оборудования (коммутаторов) и фрагментов распределенной базы данных.

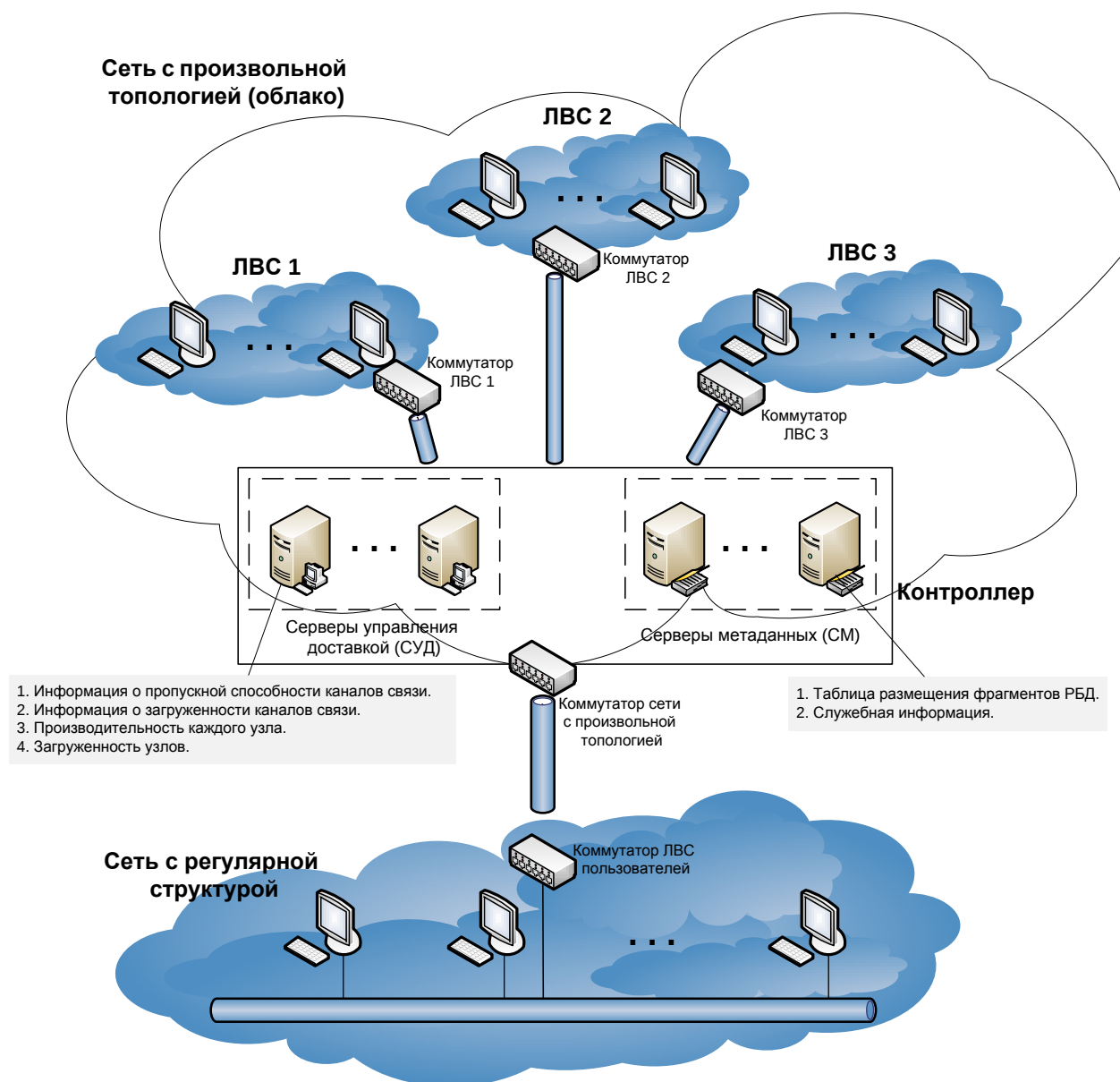


Рис. 1. Сетевая структура

Пусть кроме того имеется база данных, распределенная по узлам сети с произвольной топологией. Для упрощения расчетов примем следующие допущения:

- число серверов управления доставкой равно 1;
- число серверов метаданных равно 1;

- количество фрагментов базы данных равно 5;
- число узлов ЛВС 1 равно 2;
- число узлов ЛВС 2 равно 2;
- число узлов ЛВС 3 равно 1;
- распределение фрагментов по узлам сети задано таблицей 1.

Таблица 1

Распределение фрагментов по узлам сети

Фрагменты Узлы	Фрагмент 1	Фрагмент 2	Фрагмент 3	Фрагмент 4	Фрагмент 5
Узел 1 в ЛВС 1	+				
Узел 2 в ЛВС 1		+			
Узел 1 в ЛВС 2			+		
Узел 2 в ЛВС 2				+	
Узел 1 в ЛВС 3					+

В качестве исходных данных задаются интенсивности возникновения отказов (при этом считается, что вероятность возникновения отказов распределена по экспоненциальному закону):

$$\lambda_i, i = \overline{1, n},$$

где n – количество элементов системы, которые могут выйти из строя.

Рассмотрим данную систему как структуру, состоящую из невосстанавливаемых элементов. Для определения показателей надежности применим метод построения дерева отказов [15]. Под отказом в данном случае понимается нарушение доступа хотя бы к одному из узлов, хранящих по меньшей мере один недублированный фрагмент.

С достаточной для решаемой задачи (определение надежности той части системы, которая расположена в облаке) степенью детализации вся структура на физическом уровне показана (рис. 2).

Далее определим набор базисных состояний, которые непосредственно связаны с возникновением отказа:

- a_1 – отказ коммутатора сети с произвольной топологией;
- $a_2, a_4, a_6, a_8, a_{10}, a_{12}, a_{14}, a_{16}, a_{18}, a_{20}$ – отказы каналов связи;
- a_3 – отказ сервера управления доставкой;
- a_5 – отказ сервера метаданных;
- a_7, a_9, a_{11} – отказы коммутаторов ЛВС 1, ЛВС 2 и ЛВС 3 соответственно;

– $a_{13}, a_{15}, a_{17}, a_{19}, a_{21}$ – отказы узлов, хранящих фрагменты РБД.

Перечисленным состояниям $a_i, i = \overline{1, n}$, соответствуют вероятности возникновения отказов $q_i(t)$, которые связаны с вероятностями надежной работы элементов системы $p_i(t)$ следующим соотношением:

$$q_i(t) = 1 - p_i(t).$$

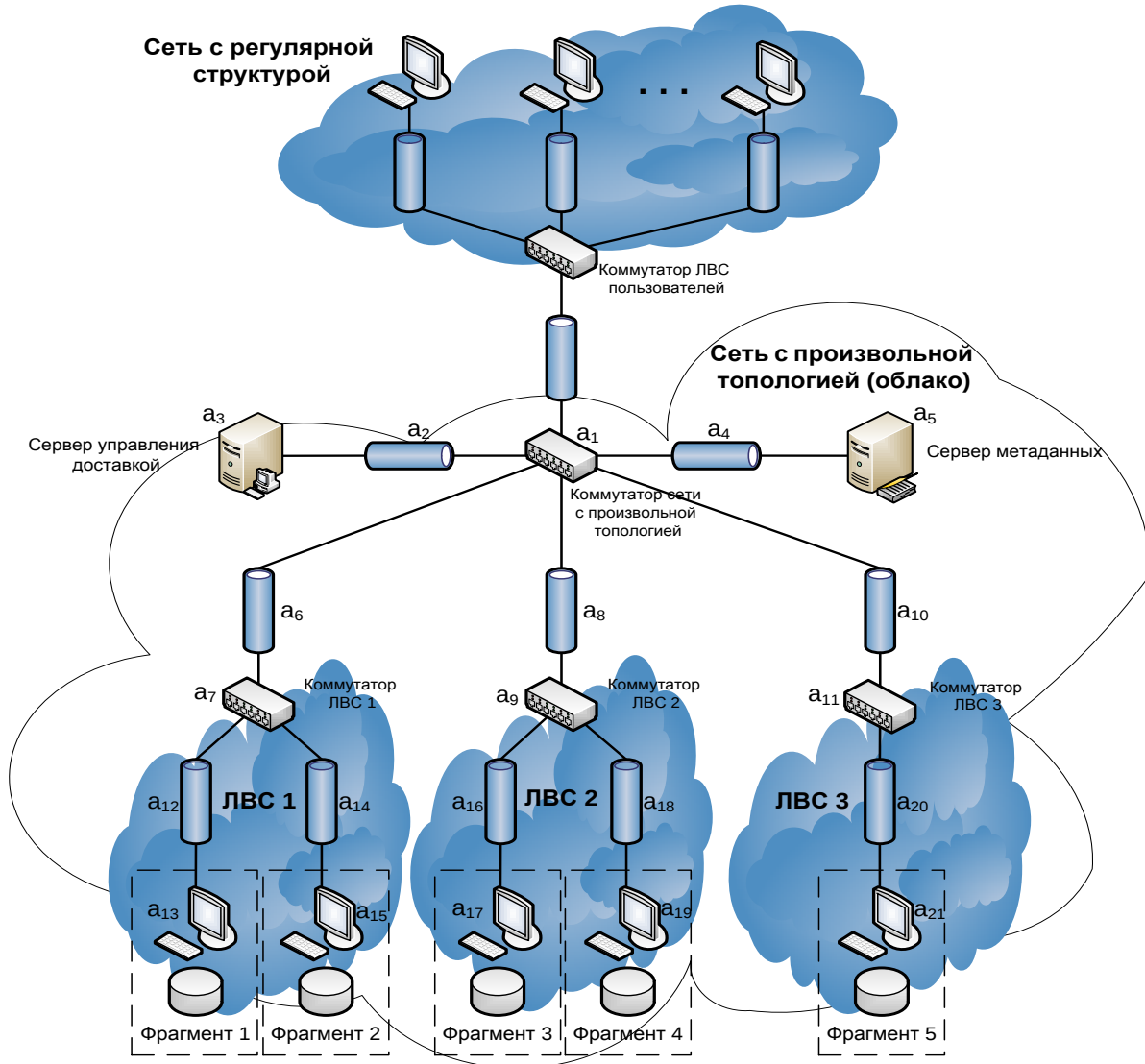


Рис. 2. Физическая структура сети

Так как в рассматриваемом варианте информационной системы вероятность возникновения отказов распределена по экспоненциальному закону, то справедливо выражение:

$$p_i(t) = e^{-\lambda_i t}, \quad i = \overline{1, n},$$

где λ_i – интенсивность возникновения отказов i -го элемента, заданная в качестве исходных данных.

На основе базисных состояний строится дерево событий (рис. 3), раскрывающее переход от базисных состояний к основному событию z_0 – отказ системы. Все рассматриваемые события связываются логическими операциями – конъюнкцией $\wedge$ (И) или дизъюнкцией $\vee$ (ИЛИ). В данной модели отказы являются независимыми событиями и наступают в произвольные моменты времени, причем возможен одновременный выход из строя нескольких элементов.

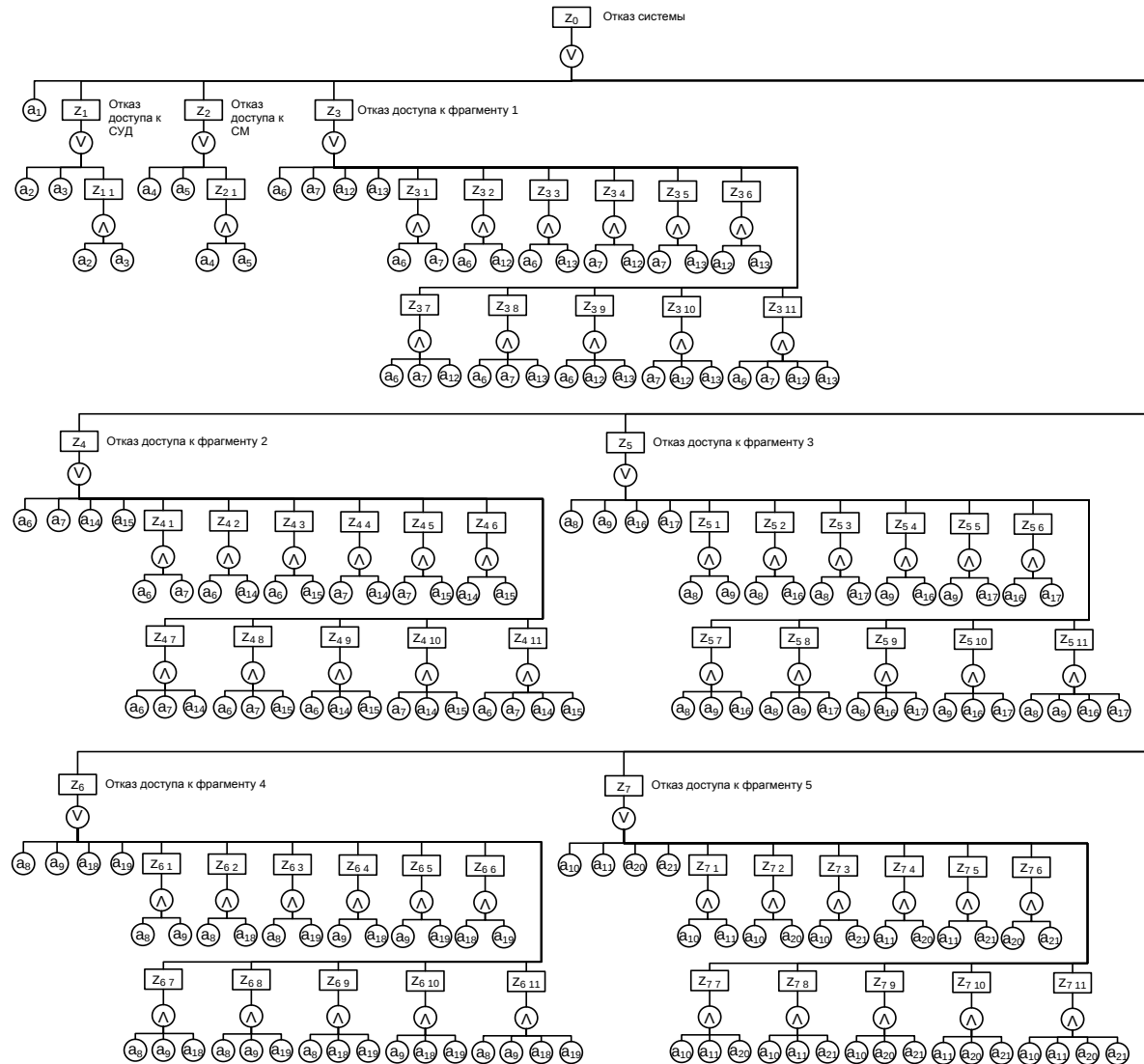


Рис. 3. Дерево событий

Затем на множестве базисных состояний выделяется множество минимальных сечений – набор состояний, которые вызывают отказ, – и строится последовательно-параллельная схема, содержащая в качестве элементов множество минимальных сечений (рис. 4).

Для данной схемы вычисляется вероятность безотказной работы на интервале $[0; t]$, причем при последовательном соединении элементов она вычисляется по формуле:

$$p_{\Sigma}(t) = \prod_{i=1}^m p_i(t),$$

где m – число последовательно соединенных элементов.

При параллельном соединении вероятность безотказной работы определяется следующим образом:

$$p_{\Sigma}(t) = 1 - q_{\Sigma}(t),$$

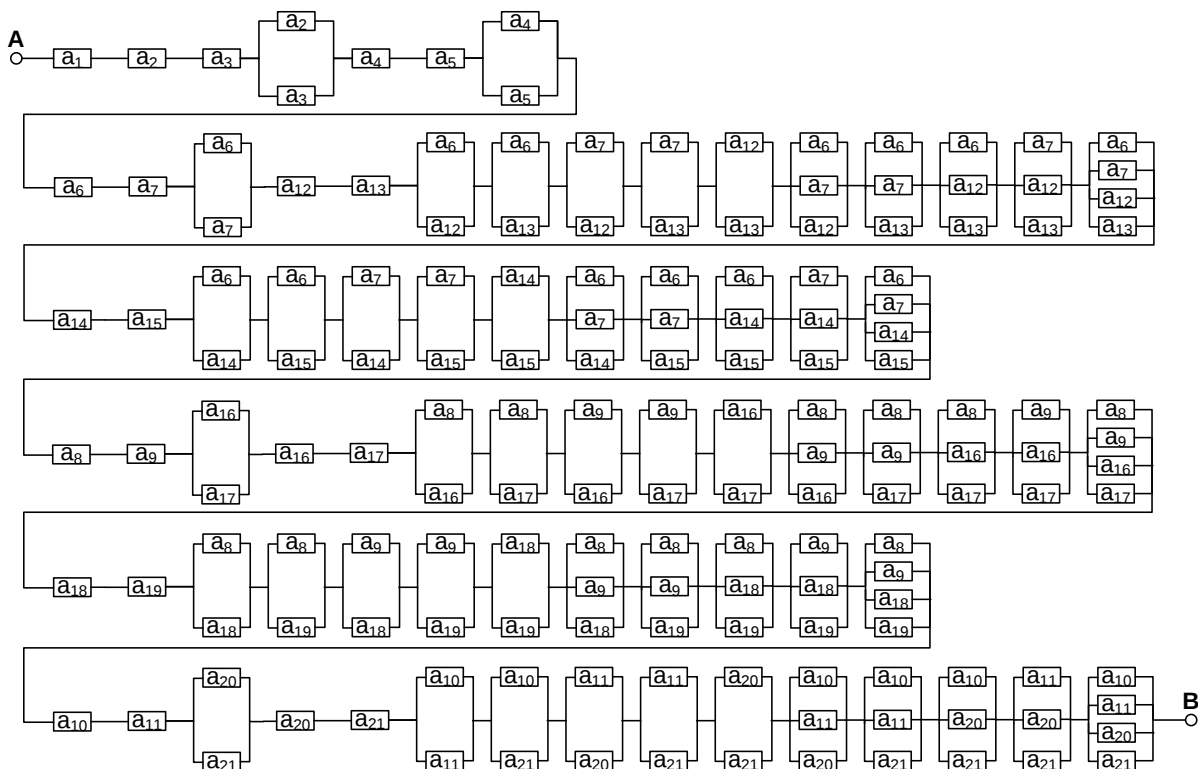


Рис. 4. Последовательно-параллельная схема

Здесь $q_{\Sigma}(t)$ – функция ненадежности, вычисляемая по формуле:

$$q_{\Sigma}(t) = \prod_{i=1}^m q_i(t) = \prod_{i=1}^m (1 - p_i(t)), \quad (1)$$

где m – число параллельно соединенных элементов.

С учетом вышеизложенного, вероятность безотказной работы исследуемой системы определяется следующим образом:

$$\begin{aligned}
P(t) = & p_1(t)p_2(t)p_3(t)(1-q_{2,3}(t))p_4(t)p_5(t)(1-q_{4,5}(t)) \times \\
& \times p_6(t)p_7(t)(1-q_{6,7}(t))p_{12}(t)p_{13}(t)(1-q_{6,12}(t))(1-q_{6,13}(t))(1-q_{7,12}(t))(1-q_{7,13}(t)) \times \\
& \times (1-q_{12,13}(t))(1-q_{6,7,12}(t))(1-q_{6,7,13}(t))(1-q_{6,12,13}(t))(1-q_{7,12,13}(t))(1-q_{6,7,12,13}(t)) \times
\end{aligned} \tag{2}$$

$$\begin{aligned}
& \times p_{14}(t)p_{15}(t)(1-q_{6,14}(t))(1-q_{6,15}(t))(1-q_{7,14}(t))(1-q_{7,15}(t)) \times \\
& \times (1-q_{14,15}(t))(1-q_{6,7,14}(t))(1-q_{6,7,15}(t))(1-q_{6,14,15}(t))(1-q_{7,14,15}(t))(1-q_{6,7,14,15}(t)) \times \\
& \times p_8(t)p_9(t)(1-q_{8,9}(t))p_{16}(t)p_{17}(t)(1-q_{8,16}(t))(1-q_{8,17}(t))(1-q_{9,16}(t))(1-q_{9,17}(t)) \times \\
& \times (1-q_{16,17}(t))(1-q_{8,9,16}(t))(1-q_{8,9,17}(t))(1-q_{8,16,17}(t))(1-q_{9,16,17}(t))(1-q_{8,9,16,17}(t)) \times \\
& \times p_{18}(t)p_{19}(t)(1-q_{8,18}(t))(1-q_{8,19}(t))(1-q_{9,18}(t))(1-q_{9,19}(t)) \times \\
& \times (1-q_{18,19}(t))(1-q_{8,9,18}(t))(1-q_{8,9,19}(t))(1-q_{8,18,19}(t))(1-q_{9,18,19}(t))(1-q_{8,9,18,19}(t)) \times \\
& \times p_{10}(t)p_{11}(t)(1-q_{10,11}(t))p_{20}(t)p_{21}(t)(1-q_{10,20}(t))(1-q_{10,21}(t))(1-q_{11,20}(t))(1-q_{11,21}(t)) \times \\
& \times (1-q_{20,21}(t))(1-q_{10,11,20}(t))(1-q_{10,11,21}(t))(1-q_{10,20,21}(t))(1-q_{11,20,21}(t))(1-q_{10,11,20,21}(t)).
\end{aligned}$$

Следуя формуле (1), выражение (2) можно записать в следующем виде:

$$\begin{aligned}
P(t) = & p_1(t)p_2(t)p_3(t)(1-(1-p_2(t))(1-p_3(t)))p_4(t)p_5(t)(1-(1-p_4(t))(1-p_5(t))) \times \\
& \times p_6(t)p_7(t)(1-(1-p_6(t))(1-p_7(t)))p_{12}(t)p_{13}(t)(1-(1-p_6(t))(1-p_{12}(t))) \times \\
& \times (1-(1-p_6(t))(1-p_{13}(t)))(1-(1-p_7(t))(1-p_{12}(t)))(1-(1-p_7(t))(1-p_{13}(t))) \times \\
& \times (1-(1-p_{12}(t))(1-p_{13}(t)))(1-(1-p_6(t))(1-p_7(t))) \times \\
& \times (1-p_{12}(t))(1-(1-p_6(t))(1-p_7(t))(1-p_{13}(t))) \times \\
& \times (1-(1-p_6(t))(1-p_{12}(t))(1-p_{13}(t)))(1-(1-p_7(t))(1-p_{12}(t))(1-p_{13}(t))) \times \\
& \times (1-(1-p_6(t))(1-p_7(t))(1-p_{12}(t))(1-p_{13}(t))) \times \\
& \times p_{14}(t)p_{15}(t)(1-(1-p_6(t))(1-p_{14}(t)))(1-(1-p_6(t))(1-p_{15}(t))) \times \\
& \times (1-(1-p_7(t))(1-p_{14}(t)))(1-(1-p_7(t))(1-p_{15}(t))) \times \\
& \times (1-(1-p_{14}(t))(1-p_{15}(t)))(1-(1-p_6(t))(1-p_7(t))(1-p_{14}(t))) \times \\
& \times (1-(1-p_6(t))(1-p_7(t))(1-p_{15}(t))) \times \\
& \times (1-(1-p_6(t))(1-p_{14}(t))(1-p_{15}(t)))(1-(1-p_7(t))(1-p_{14}(t))(1-p_{15}(t))) \times \\
& \times (1-(1-p_6(t))(1-p_7(t))(1-p_{14}(t))(1-p_{15}(t))) \times \\
& \times p_8(t)p_9(t)(1-(1-p_8(t))(1-p_9(t)))p_{16}(t)p_{17}(t)(1-(1-p_8(t))(1-p_{16}(t))) \times \\
& \times (1-(1-p_8(t))(1-p_{17}(t)))(1-(1-p_9(t))(1-p_{16}(t)))(1-(1-p_9(t))(1-p_{17}(t))) \times \\
& \times (1-(1-p_{16}(t))(1-p_{17}(t)))(1-(1-p_8(t))(1-p_9(t))(1-p_{16}(t))) \times \\
& \times (1-(1-p_8(t))(1-p_9(t))(1-p_{17}(t)))(1-(1-p_8(t))(1-p_{16}(t))(1-p_{17}(t))) \times \\
& \times (1-(1-p_9(t))(1-p_{16}(t))(1-p_{17}(t))) \times \\
& \times (1-(1-p_8(t))(1-p_9(t))(1-p_{16}(t))(1-p_{17}(t))) \times \\
& \times p_{18}(t)p_{19}(t)(1-(1-p_8(t))(1-p_{18}(t)))(1-(1-p_8(t))(1-p_{19}(t))) \times \\
& \times (1-(1-p_9(t))(1-p_{18}(t)))(1-(1-p_9(t))(1-p_{19}(t))) \times \\
& \times (1-(1-p_{18}(t))(1-p_{19}(t)))(1-(1-p_8(t))(1-p_9(t))(1-p_{18}(t))) \times
\end{aligned} \tag{3}$$

$$\begin{aligned}
& \times (1 - (1 - p_8(t))(1 - p_9(t))(1 - p_{19}(t))) (1 - (1 - p_8(t))(1 - p_{18}(t))(1 - p_{19}(t))) \times \\
& \quad \times (1 - (1 - p_9(t))(1 - p_{18}(t))(1 - p_{19}(t))) \times \\
& \quad \times (1 - (1 - p_8(t))(1 - p_9(t))(1 - p_{18}(t))(1 - p_{19}(t))) \times \\
& \times p_{10}(t) p_{11}(t) (1 - (1 - p_{10}(t))(1 - p_{11}(t))) p_{20}(t) p_{21}(t) (1 - (1 - p_{10}(t))(1 - p_{20}(t))) \times \\
& \times (1 - (1 - p_{10}(t))(1 - p_{21}(t))) (1 - (1 - p_{11}(t))(1 - p_{20}(t))) (1 - (1 - p_{11}(t))(1 - p_{21}(t))) \times \\
& \quad \times (1 - (1 - p_{20}(t))(1 - p_{21}(t))) (1 - (1 - p_{10}(t))(1 - p_{11}(t))(1 - p_{20}(t))) \times \\
& \quad \times (1 - (1 - p_{10}(t))(1 - p_{11}(t))(1 - p_{21}(t))) \times \\
& \times (1 - (1 - p_{10}(t))(1 - p_{20}(t))(1 - p_{21}(t))) (1 - (1 - p_{11}(t))(1 - p_{20}(t))(1 - p_{21}(t))) \times \\
& \quad \times (1 - (1 - p_{10}(t))(1 - p_{11}(t))(1 - p_{20}(t))(1 - p_{21}(t))).
\end{aligned}$$

Для определения функции ненадежности $F(t)$ рассматриваемой части системы необходимо использовать известное из теории вероятностей соотношение:

$$F(t) = 1 - P(t).$$

Теперь рассмотрим вариант реализации описанной выше системы, при котором для повышения надежности используется увеличение числа каналов связи. В простом случае можно использовать дублирование линии путем агрегирования линий связи.

Метод агрегирования линий связи описан в стандарте IEEE 802.3ad «Aggregation of Multiple Link Segments», который является разделом общего стандарта IEEE 802.3 [16]. Этот метод использует два и более параллельных кабелей и портов для каждой линии связи. Объединение нескольких физических линий связи в один логический канал осуществляется с помощью протокола Link Aggregation Control Protocol (LACP). При этом группа (агрегат) линий связи и портов представляется одним логическим сервисным интерфейсом с одним MAC-адресом. По протоколу LACP полные Ethernet-фреймы попеременно отсылаются по параллельным линиям связи и объединяются в приёмнике. Пропускная способность такого агрегированного канала оказывается прямо пропорциональной количеству физических линий. При отказе одной линии данные пересылаются по другой. Этот стандарт поддерживается многими производителями Ethernet-коммутаторов.

Метод резервирования, изложенный в стандарте IEEE 802.3ad, предполагает, что все агрегированные линии связи должны исходить из одного и того же коммутатора, то есть сеть должна иметь топологию звезды. Для устранения этого ограничения фирмой Nortel были предложены три модификации метода агрегирования: SMLT (Split Multi-Link Trunking), DSMLT (Distributed Split Multi-Link Trunking) и R-SMLT (Routed-SMLT).

Модификации этого метода предложены также фирмами Cisco и Adaptec, однако они не совместимы между собой и со стандартом.

Метод агрегирования используется для резервирования соединений между коммутаторами, между коммутатором и сервером, а также между двумя компьютерами. Для дублирования связи между рабочей станцией и коммутатором необходимо, чтобы оба устройства имели по два Ethernet-порта и драйвер, поддерживающий протокол LACP (IEEE 802.3ad), который предоставляет операционной системе один сетевой порт, физически состоящий из двух линий связи. При использовании 4-кратного резервирования связи между сервером и коммутатором в сервере устанавливается специальная 4-портовая Ethernet-карта с соответствующим драйвером, который заменяет 4 физических Ethernet-порта одним логическим.

Достоинством метода является увеличение пропускной способности сети, возможность добавления произвольного количества линий связи для согласования пропускной способности разных каналов, малое время восстановления после отказа. Однако для резервирования сети в целом необходимо удвоенное количество кабелей и коммутаторов, что может быть неоправданно дорого. Кроме того, практически используемые схемы агрегирования часто не соответствуют стандартам IEEE, а оборудование разных производителей может быть несовместимым [17].

Метод агрегирования в соответствии с IEEE 802.3ad обеспечивает резервирование только линий связи; коммутаторы или сетевые контроллеры подключённого к сети оборудования остаются нерезервированными. Однако некоторые фирмы (например, компания SysKonnnect) предлагают дополнительное программное обеспечение, позволяющее объединять в один логический порт несколько каналов, проходящих через разные коммутаторы, которые таким образом оказываются резервированными.

Необходимо отметить, что существуют такие варианты реализации физической структуры сети, которые не позволяют дублировать уже существующий канал связи (например, между коммутатором сети облачной структуры и коммутатором подсети, входящей в облако). В таких случаях для повышения надежности, а также пропускной способности сети можно использовать специализированное программное обеспечение, упомянутое выше, или (при наличии возможности) добавить линию связи, образующую параллельный канал посредством использования другой подсети.

Пусть в рассмотренной выше сети (рис. 2) дублируется канал связи между коммутатором сети облачной структуры и коммутатором ЛВС 2, а также добавляется связь между коммутаторами ЛВС 1 и ЛВС 2.

Тогда структуру на физическом уровне можно изобразить так, как показано на рис. 5.

Для определения показателей надежности такого варианта системы можно также применить метод построения дерева отказов. Тогда к определенному выше набору базисных состояний дополнительно вводятся:

- a_{22} – отказ канала связи между коммутатором сети облачной структуры и коммутатором ЛВС 2;

- a_{23} – отказы канала связи между коммутаторами ЛВС 1 и ЛВС 2.

Можно заметить, что размерность задачи существенно возрастает. Если данного случая построить функцию $P(t)$, то можно убедиться в том, что ее график будет лежать выше графика функции (3) на интервале $t \geq 0$. Это свидетельствует о том, что увеличение числа каналов связи (за счет дублирования линий и введения дополнительных соединений) приводит к увеличению надежности системы.

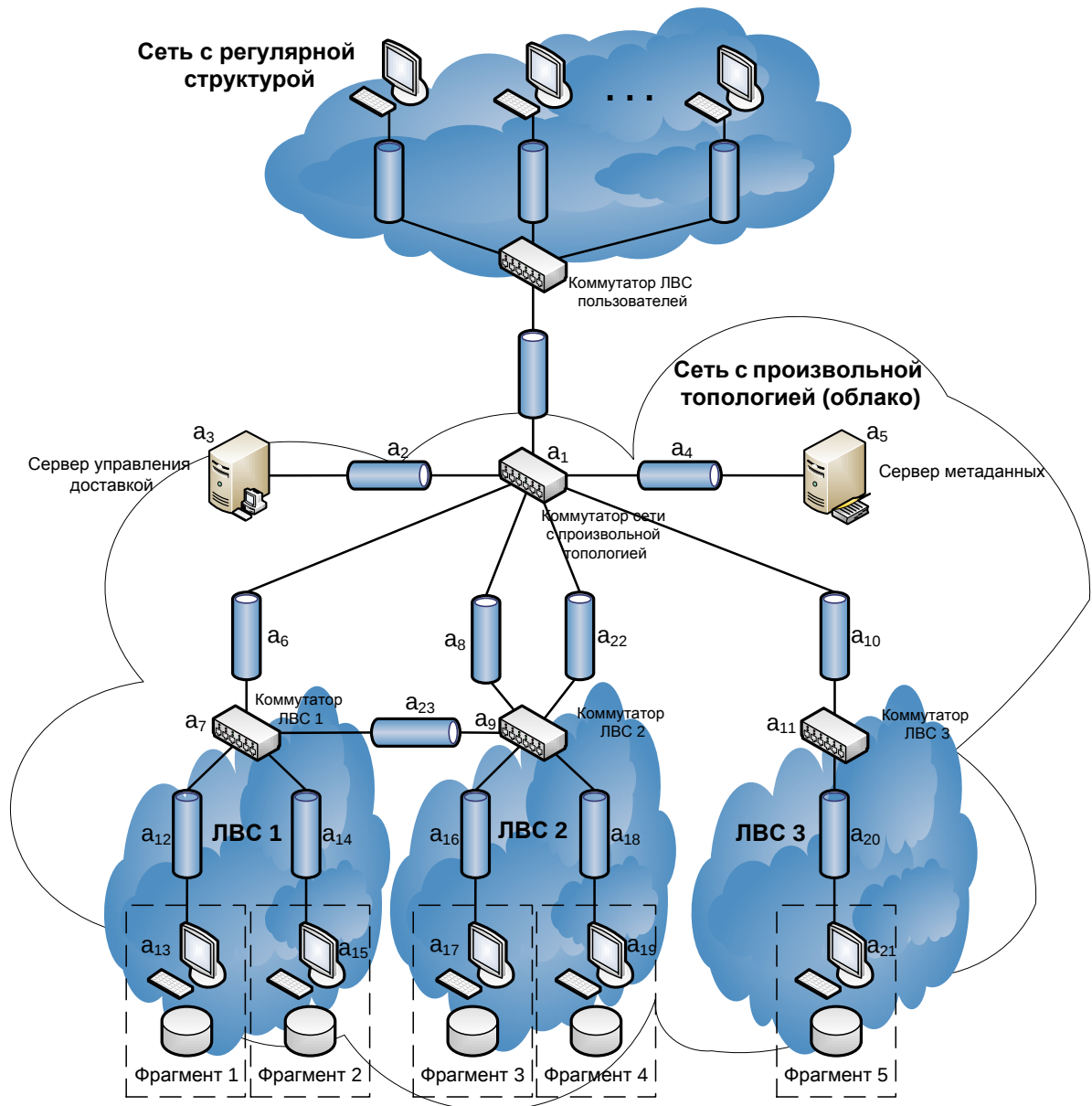


Рис. 5. Физическая структура сети с дублированием каналов

Для определения показателей надежности такого варианта системы можно также применить метод построения дерева отказов. Тогда к определенному выше набору базисных состояний дополнительно вводятся:

- a_{22} – отказ канала связи между коммутатором сети облачной структуры и коммутатором ЛВС 2;
- a_{23} – отказы канала связи между коммутаторами ЛВС 1 и ЛВС 2.

Можно заметить, что размерность задачи существенно возрастает. Если данного случая построить функцию $P(t)$, то можно убедиться в том, что ее график будет лежать выше графика функции (3) на интервале $t \geq 0$. Это свидетельствует о том, что увеличение числа каналов связи (за счет

дублирования линий и введения дополнительных соединений) приводит к увеличению надежности системы.

Кроме того, можно показать, что повышению вероятности безотказной работы способствует также дублирование сетевого оборудования (коммутаторов) и непосредственно узлов сети.

Если физическая структура системы не позволяет использовать дополнительное оборудование или каналы связи, то требованию обеспечить высокую доступность данных, хранящихся на узлах сети облачной структуры в условиях использования облачной архитектуры со случайной структурой связей подсетей (предполагающей выход одних узлов из сети и ввод новых узлов в эксплуатацию непосредственно в процессе функционирования системы) может удовлетворять только система с избыточностью данных. Это подразумевает использование репликации фрагментов РБД. Поскольку в данном варианте реализации системы не предполагается использовать аппаратные средства резервирования, предлагается репликация фрагментов на нескольких узлах хранения в системе. Узлы, содержащие одинаковые файлы, объединяются в группы. При этом группа может содержать в себе узлы, физически находящиеся в разных подсетях. Такой принцип резервирования данных будем называть сетевым зеркалированием узлов.

Под репликацией в рассматриваемом контексте понимается процесс создания нескольких копий (реплик) одного и того же фрагмента базы данных, каждая из которых хранится на отдельном узле. Управление репликами происходит централизованно через сервер метаданных, что гарантирует целостность и непротиворечивость данных.

Принято считать, что репликация является наиболее эффективным решением для обеспечения доступности данных. Это объясняется тем, что при выходе из строя одного устройства хранения, нужные данные можно получить, обратившись к другому устройству, на котором имеется копия требуемого фрагмента.

Таким образом, система оперирует несколькими репликами фрагментов, причем каждая копия находится на отдельном узле. Имеется несколько причин для такого решения, главными из которых являются:

- увеличение доступности за счет наличия независимых копий каждого фрагмента на разных узлах, и при отказе одного из них фрагмент остается доступным;
- возможность распределения нагрузки между узлами.

При увеличении количества реплик фрагмента растет доступность и скорость доступа к ним. С другой стороны, увеличение числа копий приводит к росту затрат памяти. Так возникает задача управления репликами, решение которой должно снижать необоснованные затраты.

Рекомендуемое число реплик N_i ($i = \overline{1, m}$, где m – число фрагментов), согласно [18], предлагается рассчитывать как сумму минимального числа реплик N_{min} , принимаемого из соображений обеспечения принципа достаточности данных, и верхней границы N_{add} числа дополнительных реплик.

При расчете дополнительного числа реплик будем исходить из:

- популярности фрагмента R_i , то есть количества обращений к фрагменту за определенный период времени;
- степени важности фрагмента I_i ;
- размера фрагмента T_i .

Важность каждого из параметров оценивается в баллах (соответственно w_{pop} , w_{imp} , w_{size} (экспертная оценка)) для учета их значимости при вычислении дополнительного количества реплик.

На основе вектора весовых коэффициентов и значений параметров вычисляется рекомендуемое число реплик для i -го фрагмента:

$$N_i = N_{min} + N_{add} \cdot \left(w_{pop} \frac{R_i}{R_{max}} + w_{imp} \frac{I_i}{I_{max}} + w_{size} \frac{T_i}{T_{max}} \right), \quad i = \overline{1, m},$$

где N_{add} – верхняя граница числа дополнительных реплик; R_{max} , I_{max} , T_{max} – максимальные значения параметров; w_{pop} , w_{imp} , w_{size} – весовые коэффициенты; m – число фрагментов.

Для оценки надежности такого варианта системы также можно использовать метод построения дерева отказов.

Таким образом, для повышения надежности транзакционной системы, построенной по принципу облачной структуры, можно применять различные методы (дублирование каналов связи, сетевого оборудования, узлов, репликация фрагментов РБД и др.), рассмотренные в данной статье. Однако достаточно остро стоит проблема определения показателей надежности строго аналитическими методами, которая усложняется тем, что коммутационная сеть, к анализу которой в конечном итоге сводится данная задача, является сложной структурой. При этом использование метода полного перебора становится нерациональным по причине большого количества варьируемых переменных. Поэтому актуальной становится задача построения алгоритма, позволяющего производить расчет надежности сложных параллельно-последовательных сетей без использования высокопроизводительных ЭВМ.

Литература

1. Афанасьева В.И. Разработка моделей и методов структурного резервирования программного и информационного обеспечения автоматизированных информационных систем : дис. канд. техн. наук : 05.25.05. – М., 2001. – 119 с.
2. Глушков В.М. Сети ЭВМ. – М.: Связь, 1977. – 280 с.
3. Флинт Д. Локальные сети ЭВМ: архитектура, принципы построения, реализация. – М.: Финансы и статистика, 1986. – 359 с.
4. Якубайтис Э.А. Архитектура вычислительных сетей. – М.: Статистика, 1980. – 279 с.
5. Якубайтис Э.А. Информационно-вычислительные сети. – М.: Финансы и статистика, 1984. – 232 с.
6. Якубайтис Э.А. Открытые информационные сети. – М.: Радио и связь, 1991. – 208 с.: ил.
7. Артамонов Г.Т., Тюрин В.Д. Топология сетей ЭВМ и многопроцессорных систем. – М.: Радио и связь, 1991. – 248 с.
8. Янбых Г.Ф., Столяров Б.А. Оптимизация информационно-вычислительных систем. – М.: Радио и связь, 1987. – 232 с.
9. Зайченко Ю.П., Гонга Ю.В. Структурная оптимизация сетей ЭВМ. – Киев: Техника, 1986. – 168 с.
10. Хорошевский В.Г. Инженерный анализ функционирования вычислительных машин и систем. – М.: Радио и связь, 1987. – 256 с.
11. Турута Е.Н. Обеспечение отказоустойчивости управляющих микропроцессорных систем путем перераспределения задач отказавших модулей. Системы управления информационных сетей. – М.: Наука, 1983. – С. 187-197.
12. Турута Е.Н., Аскеров Ч.И., Фургина Л.А. Распределение задач с целью обеспечения отказоустойчивости многопроцессорных вычислительных систем. Сетевые протоколы и управление в распределенных вычислительных системах. – М.: Наука, 1986. – С. 168-173.
13. Кондратьев К.О. Метод динамического перераспределения управляющих программ в распределенном вычислительном комплексе. Автоматика и вычислительная техника. – 1987. – Т6. – С. 15-18.
14. Мамиконов А.Г., Кульба В.В. Синтез оптимальных модульных систем обработки данных. – М.: Наука, 1986. – 275 с.
15. Рябинин И.А. Надёжность и безопасность структурно-сложных систем. – СПб.: Политехника, 2000. – 248 с.
16. IEEE Std. 802.3. IEEE standard for information technology – Telecommunications and information exchange between systems – Local and metropolitan area network – Specific requirements. Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications. – IEEE Computer Society. 2005. Section 1–5. 2628 p.
17. Денисенко В.В. Аппаратное резервирование в промышленной автоматизации // В записную книжку инженера. – 2008. – №4. – С. 92-100.
18. Лукьянов Н.М. Разработка архитектуры и методов организации слабосвязанных архивных систем для автоматизации проектирования : дис. канд. техн. наук : 05.13.12. – СПб.: СПбГУ ИТМО, 2011. – 85 с.

346428, г. Новочеркасск, ул. Просвещения, 132, т.255-4-33, e-mail: vtl88@mail.ru

УДК 378.091

АЛГОРИТМ ПОСТРОЕНИЯ СКОРИНГОВЫХ МОДЕЛЕЙ НА ОСНОВЕ ДЕРЕВА РЕШЕНИЙ

Гринченков Д.В., Горбушин А.А., Горбушин Д.А.

Южно-Российский государственный технический университет (НПИ), г. Новочеркасск

Рассмотрен один из подходов к формированию практико-ориентированных навыков на стыке применения профессиональных знаний и знаний в области информационных технологий.

Скоринговые модели в розничном кредитовании следует использовать на различных этапах работы с клиентом, начиная с получения первой заявки от заемщиков, через организацию работы с текущими заемщиками (сопровождение кредита, увеличение лимита кредитования, последующие выдачи кредитов, работа с просроченной задолженностью и т.д.), и заканчивая повышением лояльности клиентов к кредитной организации.

Существует алгоритм построения скоринговых моделей в виде дерева решений, примечательной особенностью которого является использование композиции статистического и экспертного подходов, который необходим в случаях недостаточности и ненадежности статистических данных, а также когда необходимо учесть существенное изменение экономической ситуации на рынке, не дожидаясь изменений в кредитном портфеле.

Данными на входе алгоритма являются статистические данные и знания эксперта, формализованные в виде правил следующей структуры:

```
Если имя_атрибута { { <= | > } значение | in {значения} }
[ И имя_атрибута { { <= | > } значение | in {значения} } ]
...
[ И имя_атрибута { { <= | > } значение | in {значения} } ]
то Кредитный_риск = метка_класса.
```

Правила эксперта обладают приоритетом, приоритет снижается от первого к последнему правилу. Таким образом, если встречается правило, противоречащее ранее примененному, то оно пропускается и в построении дерева не участвует.

На выходе алгоритма – класс, к которому принадлежит рассматриваемый заемщик (например, заемщик с низким кредитным риском, средним и высоким). В зависимости от кредитной политики каждый банк может по-своему определить, какого заемщика он считает «плохим», «средним» и «хорошим», на какое число групп следует разделить заемщиков.

Алгоритм построения дерева решений представлен на рис. 1.

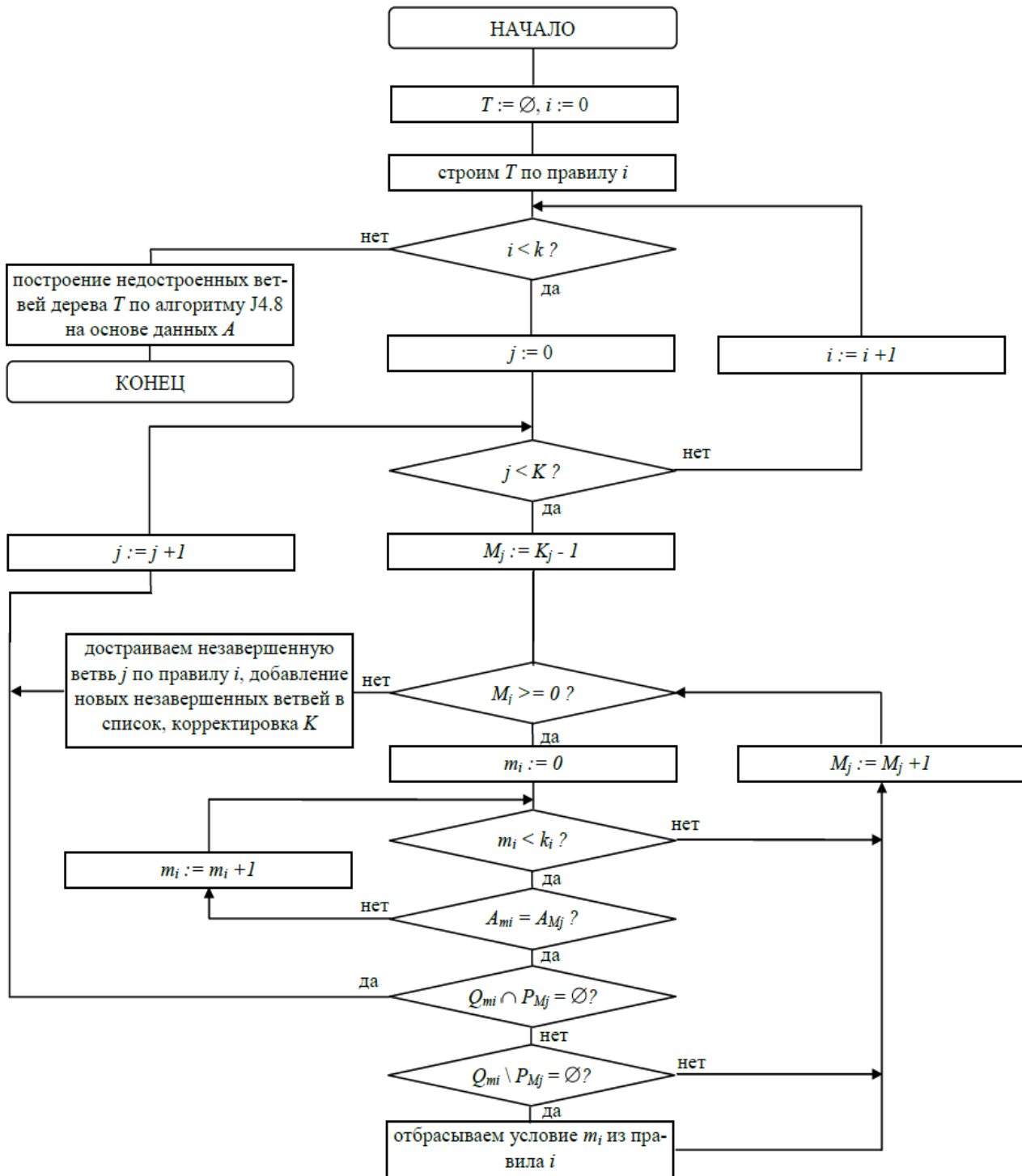


Рис. 1. Алгоритм построения скоринговой модели в виде дерева решений по правилам эксперта и данным

В рисунке использованы следующие обозначения: A – некоторое обучающее множество, содержащее объекты, каждый из которых характеризуется атрибутами, причем один из них указывает на принадлежность объекта к определенному классу; T – строящееся дерево; k – количество правил эксперта, используемых для построения дерева, i – индекс, исполь-

зуемый для обхода списка правил ($0 \leq i \leq k$); K – количество незавершенных ветвей дерева T , j – индекс, используемый для обхода списка незавершенных ветвей дерева ($0 \leq j \leq k$); m_i – индекс, используемый для обхода условий i -го правила ($0 \leq m_i \leq k_i$), k_i – количество условий в i -м правиле; M_j – индекс, используемый для обхода j -й незавершенной ветви дерева T ($0 \leq M_j \leq K_j$), K_j – количество узлов в j -й незавершенной ветви; Q_{mi} – множество значений атрибута условия A_{mi} в i -м правиле; P_{M_j} – множество значений атрибута A_{M_j} проверки узла M_j в незавершенной ветви j дерева T . Процесс построения дерева происходит сверху вниз, сначала по правилам эксперта, затем по данным с использованием алгоритма построения дерева решений J4.8. На первом шаге имеем пустое дерево (есть только корень) и исходное множество данных, ассоциированное с корнем.

После окончания алгоритма незавершенные ветви достраиваются на основе статистических данных по вышеупомянутому алгоритму J4.8. Алгоритм J4.8 реализован в библиотеке Weka (Waikato Environment for Knowledge Analysis), разработанной Университетом Waikato (Новая Зеландия). Алгоритм J4.8 использует следующий теоретико-информационный критерий: выбирается проверка по такому из атрибутов, который дает максимум отношения информационного выигрыша при разбиении по нему к оценке потенциальной информации, получаемой при разбиении множества примеров на n подмножеств, где n – число потомков узла.

На основе алгоритма построения скоринговых моделей в виде дерева решений с использованием композиции статистического и экспертного подходов и метода иерархического синтеза разработан метод построения «коллективных» скоринговых моделей, который позволяет объединить опыт и знания экспертов кредитной организации и устранить противоречия в их суждениях.

Будем считать, что эксперты используют один и тот же набор кредитных историй, атрибутов и общий критерий ветвления в алгоритме построения дерева решений по имеющимся данным, что и характерно для одной кредитной организации. Модели отличаются только набором правил, используемых экспертами, и их приоритетом.

Пусть $E_1, E_2, \dots, E_n$ – множество экспертов, $P_1, P_2, \dots, P_k$ – используемые ими правила. Для ранжирования правил относительно цели используется следующий метод, основанный на иерархическом синтезе.

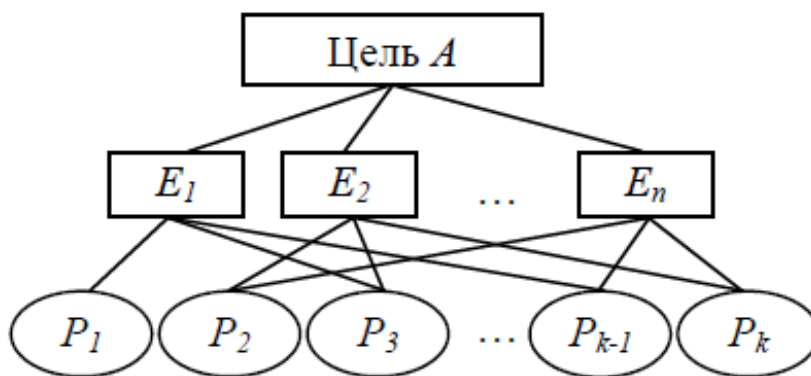


Рис. 2. Иерархическая схема задачи

Шаг 1. Структурируется исходная задача как иерархия целей. Иерархия целей строится следующим образом: верхний уровень (цель A) соответствует нахождению оптимального ранжирования правил, средний уровень соответствует системам предпочтений экспертов $E_1, E_2, \dots, E_n$, а нижний уровень – всем используемым экспертами правил $P_1, P_2, \dots, P_k$ (рис. 2). Каждый элемент среднего уровня соединен с элементом верхнего уровня, но не всегда с каждым элементом нижнего уровня.

Шаг 2. На основе иерархической структуры строится бинарная $(k \times n)$ матрица B соответствий между экспертами и правилами. Матрица B содержит элементы $b_{ij} = 0, 1$: если правило P_i используется для построения модели экспертом E_j , то $b_{ij} = 1$, в противном случае

Шаг 3. Для каждого элемента иерархии строятся матрицы парных сравнений элементов иерархии следующих уровней: $[A]$ и $[E_i]$, $i = 1, 2, \dots, n$. Для установления относительной важности элементов иерархии следует воспользоваться шкалой отношений, приведенной в таблице 1.

С использованием указанной шкалы сравниваются два эксперта с точки зрения достижения цели A , расположенной на вышележащем уровне иерархии, и указывается число в интервале от 1 до 9 или обратное значение. Для этого в иерархии выделяют элементы двух типов: элементы-родители и элементы-потомки. Элементы-потомки воздействуют на соответствующие элементы вышестоящего уровня иерархии, которые являются по отношению к первым элементами-родителями. Матрицы парных сравнений строятся для всех элементов-потомков, относящихся к определенному родителю.

Шкала отношений

Степень значимости	Характер значимости	Объяснение
1	Одинаковая значимость	Два действия вносят одинаковый вклад в достижение цели
3	Некоторое преобладание значимости одного действия над другим	Существуют соображения в пользу предпочтения одного из действий, однако эти соображения недостаточно убедительны
5	Существенная или сильная значимость	Имеются надежные данные или логические суждения для того, чтобы показать предпочтительность одного из действий
7	Очевидная или очень сильная значимость	Убедительное свидетельство в пользу одного действия перед другим
9	Абсолютная значимость	Свидетельства в пользу предпочтения одного действия перед другим в высшей степени убедительны

Парные сравнения производятся в терминах доминирования одного элемента над другим в соответствии со шкалой отношений. При проведении парных сравнений следует отвечать на вопрос: какой из двух сравниваемых элементов важнее или имеет более высокую степень воздействия. Если элемент E_1 доминирует над элементом E_2 , то клетка матрицы, соответствующая строке E_1 и столбцу E_2 , заполняется числом, большим единицы, а клетка, соответствующая строке E_2 и столбцу E_1 заполняется обратным к нему числом.

Матрица парных сравнений правил каждого из экспертов строится по следующему алгоритму: каждое последующее правило из списка правил эксперта имеет приоритет на $\frac{9-1}{k-1} = \frac{8}{k-1}$ ниже, чем предыдущее, где k – количество используемых экспертом правил, т.е.

$$E = \begin{matrix} P_1^i \\ P_2^i \\ \dots \\ P_k^i \end{matrix} \begin{bmatrix} 1 & 1 + \frac{8}{k-1} & \dots & 1 + \frac{8}{k-1} k - 1 \\ \frac{1}{1 + \frac{8}{k-1}} & 1 & \dots & 1 + \frac{8}{k-1} k - 2 \\ \dots & \dots & \dots & \dots \\ \frac{1}{1 + \frac{8}{k-1} k - 1} & \frac{1}{1 + \frac{8}{k-1} k - 2} & \dots & 1 \end{bmatrix}$$

Матрица парных сравнений экспертов формируется в зависимости от значимости экспертов в кредитном учреждении согласно шкале отношений.

Шаг 4. Для построенных матриц находятся максимальные собственные значения $\lambda_i, i = 0, 1, \dots, n$ и главные собственные векторы $W_A, W_E, i = 0, 1, \dots, n$, элементы которых считаются равными приоритетам соответствующих элементов следующего уровня иерархии.

Шаг 5. Строится матрица D

$$D = \begin{matrix} & P_1 & P_2 & \dots & P_k \\ \begin{matrix} P_1 \\ P_2 \\ \dots \\ P_k \end{matrix} & \begin{bmatrix} d_{11} & d_{12} & \dots & d_{1n} \\ d_{21} & d_{22} & \dots & d_{2n} \\ \dots & \dots & \dots & \dots \\ d_{k1} & d_{k2} & \dots & d_{kn} \end{bmatrix} \end{matrix}$$

где $\{d_{ij}\}$ представляют векторы приоритетов правил относительно эксперта E_j . При этом если правило P_i не оценивается экспертом E_j , то в матрице D соответствующее значение $d_{ij} = 0$.

Шаг 6. Для повышения приоритета правил, образующих большие группы, и снижение приоритета правил в группах с их относительно небольшим числом, используется следующий структурный критерий, отображаемый диагональной матрицей L (имеется в виду, что группу определяют правила, используемые экспертом E_j , или иначе являющиеся «потомками» по отношению к эксперту E_j в иерархической схеме задачи):

$$L = \begin{bmatrix} r_1 / N & 0 & \dots & 0 \\ 0 & r_2 / N & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & r_n / N \end{bmatrix}$$

где r_j – число правил, используемых экспертом E_j ; $N = \sum_{j=1}^k r_j$ – количество всех используемых экспертами правил.

Шаг 7. Далее вычисляется ненормированный вектор приоритетов правил относительно цели A :

$$X = DLW_A.$$

Для нормирования значений вектора приоритетов правил вычисляется нормирующий множитель $K = \left(\sum_{i=1}^k x_i \right)^{-1}$,

где $x_i, i = 1, 2, \dots, k$ – значения координат ненормированного вектора приоритетов правил. Нормированный вектор приоритетов правил принять равным

$$W = K \cdot X.$$

Используя найденный вектор приоритетов W , строим «коллективную» скоринговую модель в виде дерева решений следующим образом: сначала дерево решений строится на основе множества правил $P_1, P_2, \dots, P_k$, упорядоченного в соответствии с вектором приоритетов W , причем, если $w_k = 0$, то соответствующее правило для построения скоринговой модели не используется, затем дерево достраивается на основе имеющихся статистических данных с использованием выше упомянутого алгоритма J4.8.

Количество используемых правил для построения коллективной модели можно определить как средневзвешенное количество правил каждого из экспертов, округленное к большему.

С помощью данных методов и алгоритмов построения скоринговых моделей возможна разработка программного обеспечения, которое позволит строить модели в виде дерева решений на основе статистической информации и знаний экспертов в области кредитования.

Литература

1. Буди́на Е.С. Математические и инструментальные методы оценки рисков в розничном кредитовании на основе композиции статистического и экспертного подходов // Дисс. на соискание ученой степени кандидата экономических наук. – Пермь, 2010.
2. Семенов Д.А. Управление процессом кредитования физических лиц как задача стохастической оптимизации. – М.
3. Шакирова А.Р. Скоринг как метод оценки риска неплатежеспособности заемщика. – Уфа
4. Готовчиков И.Ф. Методы снижения асимметричности информации от кредитных историй заемщиков // Оперативное управление и стратегический менеджмент в коммерческом банке. – 2003.
5. Гусева А. Кредитные бюро и кредитный скоринг // Банки и технологии. – 2004.

УДК 378.091

ПРОГРАММНЫЕ ПРОДУКТЫ ДЛЯ АНАЛИЗА ТОНАЛЬНОСТИ ТЕКСТА, ТЕКУЩЕЕ СОСТОЯНИЕ И РЕАЛИЗУЕМЫЕ ВОЗМОЖНОСТИ

Гринченков Д.В., Горбушин Д.А., Горбушин А.А.

Южно-Российский государственный технический университет (НПИ), г. Новочеркасск

Рассматриваются существующие подходы к анализу эмоциональной составляющей коммуникации в системах обработки текстовой информации. Проводится сравнительный анализ программных продуктов предназначенных для решения данной задачи, обосновывается ее актуальность.

Совсем недавно аналитики IDC прогнозировали, что новая доминирующая платформа IT-отрасли, построенная на технологиях мобильных

устройств, облачных сервисов, социальных сетей и аналитики «больших данных», начнет превращаться в магистральное направление. Сегодня расходы на эти технологии растут примерно на 18% в год, ожидается, что они принесут не менее 80% общего роста ИТ-расходов в период до 2020г.

Развитие аналитики происходит в направлении анализа данных в реальном времени с целью прогнозирования будущих событий. Происходит переход от анализа простых структурированных данных, анализируемых отдельными пользователями, к анализу сложной, часто неструктурированной информации (текст, видео и т.д.), получаемой от множества систем.

Одной из важных частей технологии извлечения знаний и аналитической обработки текстовой информации является эмотивная функция коммуникации. Ее пытаются использовать в автоматических системах оценки и сравнения объектов, например, новых продуктов и брендов известных компаний, для выявления отношения людей к событиям в политической жизни страны и т.п.

Эмоциональная составляющая коммуникации пока не столь активно применяется в системах обработки текстовой информации не только ввиду трудностей выделения «нужной» (т.е. относящейся к рассматриваемому объекту) эмоциональной лексики в текстах, но и сложности определения самого эмотивного пространства, количества и состава его измерений.

Эмоциональная оценка обычно выражается не абстрактно, а относительно какого-либо объекта, о котором идёт речь в тексте. Такой объект принято называть объектом тональности. Носителем выраженной в тексте эмоциональной оценки также обычно является вполне определённое лицо, в общем случае это автор текста. Однако если автор текста ссылается на чьё-нибудь мнение или цитирует высказывание другого человека, то носителем эмоциональной оценки (субъектом тональности) будет тот, на чьё мнение ссылаются.

Таким образом, тональность высказывания определяется тремя компонентами: субъектом тональности (кто высказал оценку), объектом тональности (о ком или о чём высказана оценка) и тональной оценкой (как оценили).

В современных системах автоматического определения эмоциональной оценки текста чаще всего используется одномерное эмотивное пространство: позитив-негатив, то есть, хорошо-плохо. Однако известны успешные случаи использования и многомерных пространств [1].

Обобщенный алгоритм определения эмоциональной составляющей текста состоит из трех этапов (рис. 1). На первом этапе обработки текста морфологический анализатор определяет, какие слова в нём представлены,

и в каких формах. После этого слова из текста, приведённые к словарной форме, сопоставляются с тональными словарями. Слова, обнаруженные в словарях отрицательных слов, помечаются как отрицательные (обозначены жирным шрифтом), слова, найденные в словарях положительных слов – как положительные (обозначены курсивом).

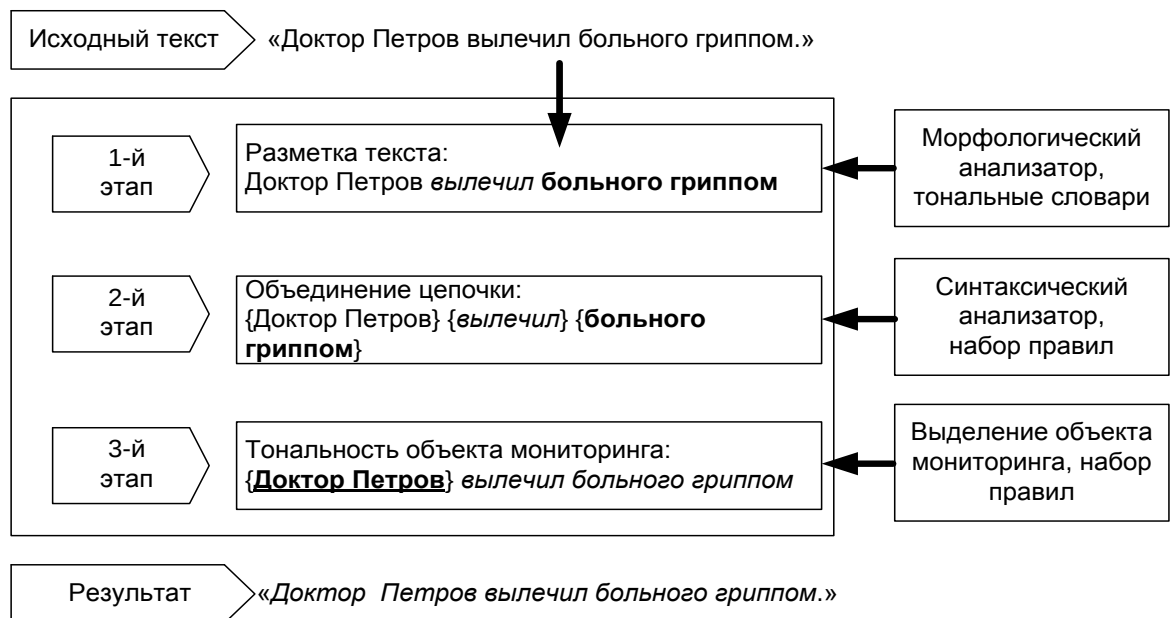


Рис. 1. Этапы определения тональности предложения

На втором этапе синтаксический анализатор объединяет связанные друг с другом слова в цепочки. Тональность каждой цепочки вычисляется специальными правилами на основании «словарной» тональности слов, входящих в эту цепочку. Например, два отрицательных существительных больной и грипп порождают отрицательную цепочку больной гриппом, а потенциально положительное существительное доктор объединяется с нейтральным именем Петров в потенциально положительную цепочку доктор Петров.

На последнем этапе в каждом предложении текста система пытается найти объект тональности данного предложения. Если он не был задан изначально в поисковом запросе, на роль объекта тональности выбирается цепочка, содержащее имя человека, а в случае её отсутствия – любое наименование человека. В примере на рис.1 объектом тональности стал доктор Петров (показано жирным шрифтом с подчеркиванием). Тональность предложения считается на основании специальных правил исходя из тональности цепочек слов, выделенных в предложении. Так, есть правило, которое гласит, что сочетание позитивного глагола вылечить с негативной цепочкой (в данном случае больной гриппом) приписывает позитив подлежащему глагола (в примере – доктору Петрову).

Можно выделить несколько систем-анализаторов тональности текста, работающих по схожему алгоритму, используя одномерное эмотивное пространство:

1. Система извлечения знаний из документов «Аналитический курьер» [6].
2. Система управления досье «X-Files»[6].
3. Проект «ВААЛ», позволяющий прогнозировать эффект неосознаваемого воздействия текстов на массовую аудиторию[8].
4. Программа для поиска фактов в тексте «RCO Fact Extractor Desktop»[9].
5. Система мониторинга социальных сетей, блогов и СМИ «Constrim DataRetriever» [7].
6. Универсальная лингвистическая платформа для приложений «ABBY COMPRENO».

Рассмотрим каждую из этих систем подробнее.

1. Система «Аналитический курьер». С помощью различного инструментария семантического анализа текста система "Аналитический курьер" позволяет извлекать знания из хранилищ текстовой информации. Она производит мониторинг контента потока и архивов документов и строит их аналитическое представление. Система определяет негативный и позитивный сантимерт относительно выделенных или заданных объектов тональности, с учётом его силы. Определение тональности является одной из функций системы извлечения знаний из документов. Пример работы Web-интерфейса программы представлен на рис. 2.

The screenshot shows the web interface of the 'Аналитический курьер' system. At the top left is the logo of 'Тесо' (Teso) with the text 'Компания «Ай-Тесо»'. The title 'Компонент определения тональности текста' is centered. On the top right are icons for a grid and a question mark. Below the title is a text input area with the placeholder 'Введите текст (не более 2000 символов)' and a character count '(длина текста: 181 символов)'. The input area contains two lines of text: 'Позитивная тональность: И.Иванов указал на необходимость совершенствования правоохранительных органов.' and 'Негативная тональность: И.Иванов обвиняется в нецелевом использовании средств.' Below the input area are two buttons: 'Выполнить' (Execute) and 'Очистить' (Clear). Below the buttons is a section titled 'Результат' (Result). The result section contains two lines of text: 'Позитивная тональность: И.Иванов указал на необходимость совершенствования правоохранительных органов.' and 'Негативная тональность: И.Иванов обвиняется в нецелевом использовании средств.' To the right of the result section is a vertical sidebar with the text 'Л е г е н д а' (Legend) and a double arrow icon.

Рис. 2. Компонент определения тональности текста

В системе реализованы следующие функции обработки текста (выполняемые автоматически):

- многоязычный поиск (формальный и на естественном языке), семантический поиск с использованием современного тезауруса русского, английского и других языков;
- автоматическое аннотирование статей;
- определение тональной окраски документов по отношению к отдельным объектам;
- автоматическое выделение тематических групп внутри коллекции документов (кластерный анализ);
- выявление ключевых тем коллекции документов, построение их взаимосвязей в виде семантической сети;
- построение дайджеста (обзора) по каждому объекту из документа;
- частотный анализ рубрик и публикаций, многомерная аналитическая обработка данных, погружение в документы через ячейку частотного распределения;
- администрирование процесса выделения знаний и доставки новой информации пользователям;
- параллельная обработка разнородной неструктурированной информации из различных источников.

2. Система управления досье «X-Files». Данная система разрабатывается совместно с системой «Аналитический курьер». Она представляет собой программный комплекс, обеспечивающий на основе фактографической информации накопление досье и аналитическую обработку содержащейся в ней информации. Система используется для выделения, накопления и анализа атрибутивной информации об объектах и семантических связях между персонами, организациями и другими объектами с использованием экспертов и семантико-лингвистических моделей обработки текстов. Основные возможности системы:

- выработка и хранение гипотез о вероятных связях объектов при отсутствии фактов об этих связях.
- высокопроизводительная аналитическая обработка большого числа фактов (более 100 000). Поиск кратчайших и эффективных связей объектов с включением связей-гипотез; анализ динамики фактов;
- хранение фактов, полученных как в источниках учетной, текстовой, так и видео - информации;
- совместная аналитическая обработка фактов, представленных на разных языках;

– ассоциативный поиск фактов за счет самообучения правил идентификации объектов (аббревиатуры, SMS, E-mail, телефон, адрес, и т.д.), что позволяет программам-автоматам находить больше фактов;

– идентификация тонально окрашенных фактов и включение их в досье объектов.

Так же данный алгоритм применяется в целях составления семантической карты взаимосвязей. Например, наиболее обсуждаемых тем в сообщениях СМИ за декабрь 2011 года по запросу «Выборы» (рис. 3).

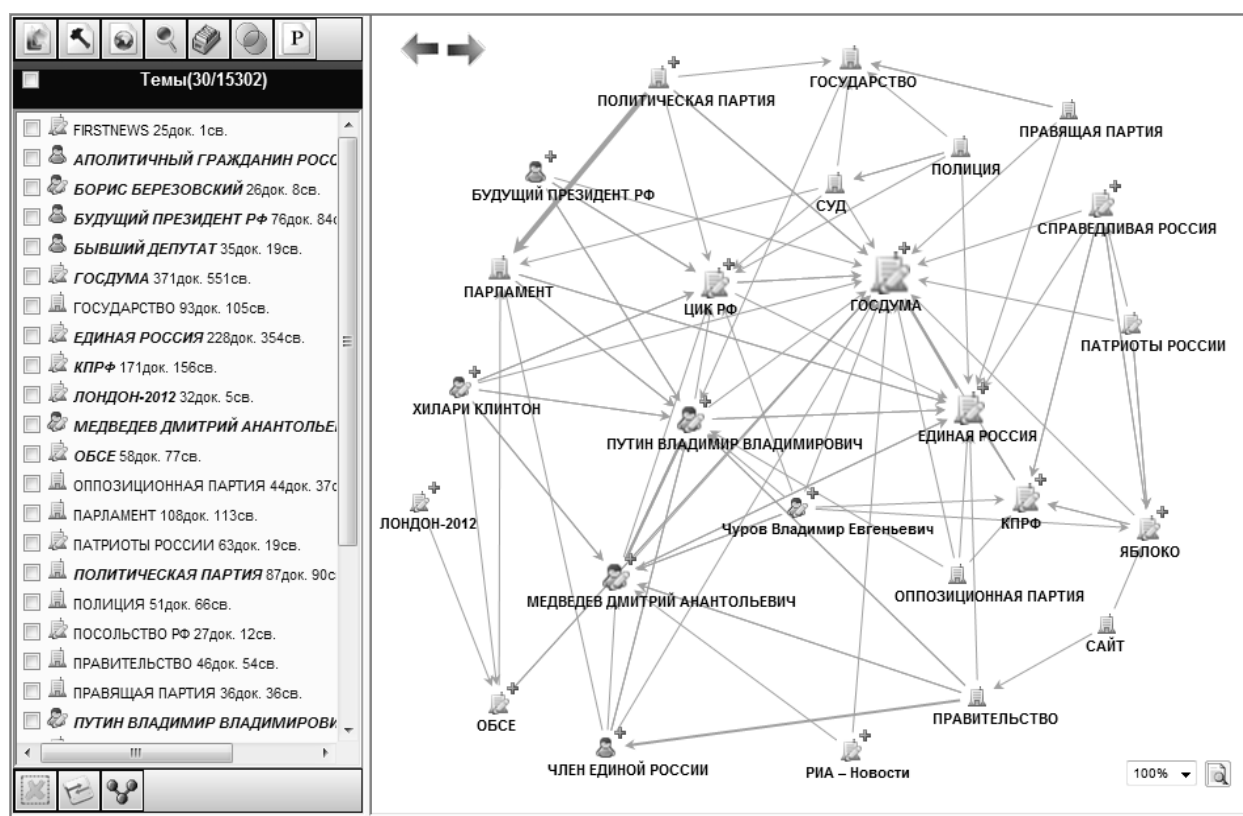


Рис. 3. Семантическая карта взаимодействия

3. Проект «ВААЛ». Проект «ВААЛ» разработан группой специалистов под руководством Дымшица Михаила. Помимо алгоритма одномерного эмотивного пространства, в системе применяется алгоритм анализа сочетания звуков. Система реализована в виде набора DLL-библиотек, которые подключаются к наиболее популярному текстовому процессору Word for Windows и в главном меню появляется новый пункт. Такой способ реализации позволяет сохранить для пользователя привычную удобную среду создания документов и максимально облегчает освоение системы «ВААЛ». Основные возможности системы:

- оценка неосознаваемого эмоционального воздействия фонетической структуры текстов и отдельных слов на подсознание человека;
- генерация слов с заданными фоносемантическими характеристиками;
- оценка неосознаваемого эмоционального воздействия фонетической структуры текстов на подсознание человека;
- оценка звуко-цветовых характеристик слов и текстов;
- позволяет производить словарный анализ текстов;
- осуществление полноценного контент-анализ текста по большому числу специально составленных встроенных категорий и категорий, задаваемых самим пользователем;
- позволяет производить выделение тем, затрагиваемых в текстах, и осуществлять на основе этого автоматическую категоризацию;
- позволяет производить эмоционально-лексический анализ текстов;
- позволяет производить вторичный анализ данных путем их визуализации, факторного и корреляционного анализа.

Система обладает возможностью анализа большого объема текста. На рис. 4 приведен пример семантического анализа рассказа И.С. Тургенева – «Муму».

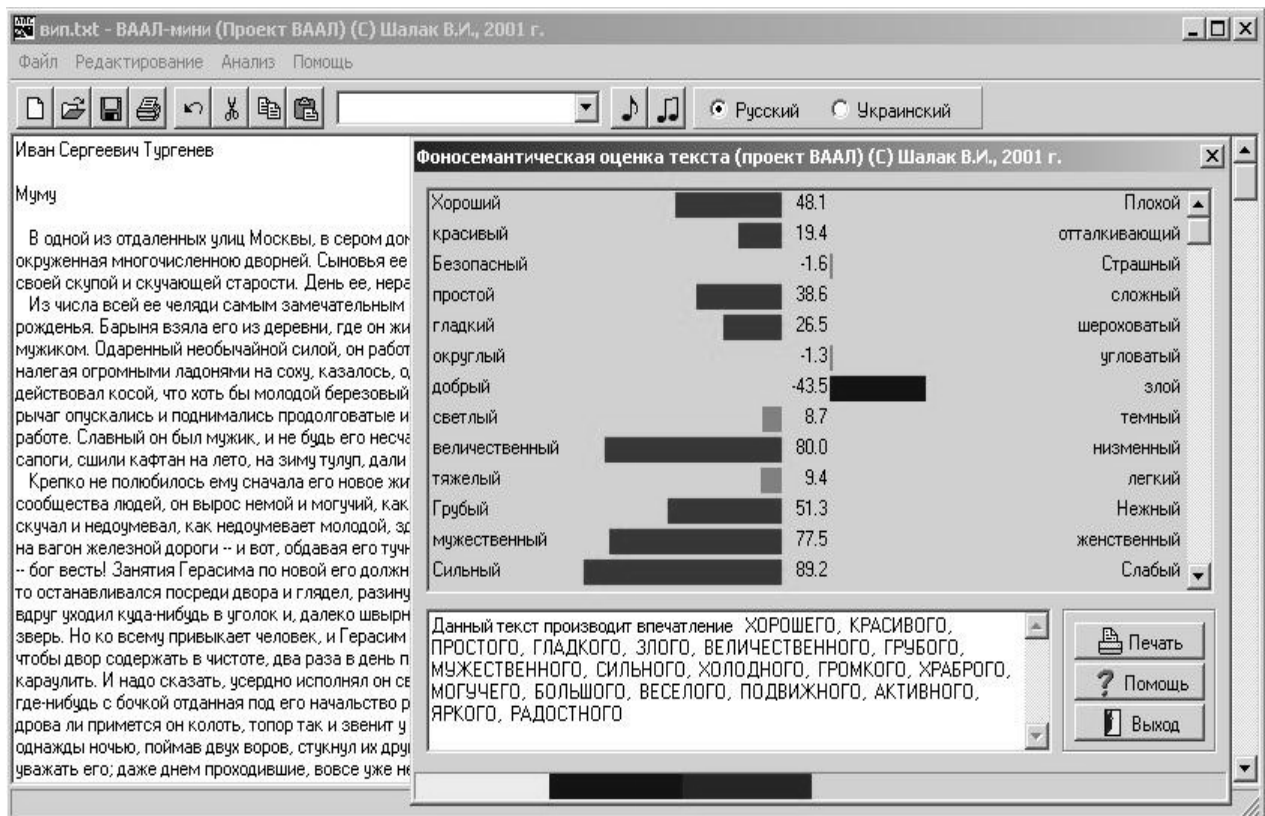


Рис. 4. Семантический анализ текста

4. Программный продукт «RCO Fact Extractor Desktop». Компанией «RCO» представлена лингвистическая модель для компьютерного анализа тональности публикаций СМИ, в основе которой лежит принцип фильтрации и разбора пропозиций. Технологически отбор пропозиций происходит как поиск в сети синтактико-семантических отношений (результат синтаксического анализа предложения) всех подсетей, аналогичных заданным лингвистическим схемам-графам. При нахождении подсети, удовлетворяющей заданной схеме, происходит отбор тех ее узлов, которые соответствуют важным для оценки тональности участникам ситуации, а также связанных с ними узлов, содержащих дополнительных информацию (частицы, наречия, прилагательные, служебные глаголы).

Программный продукт, использующий данные алгоритмы, обладает следующими возможностями:

- аналитическая обработка текста на русском языке и выявления фактов различного типа, связанных с заданными объектами – персонами и организациями;
- решение задач из области компьютерной разведки, требующих высокоточного поиска информации;
- автоматический подбор материала к досье на целевой объект или же мониторинг определенных сторон его активности, освещаемых, например, в СМИ;
- поиск не только определенных фрагментов текста, в которых говорится, например, о поездках персоны, ее встречах, заключении договоров, сделках купли-продажи, но и определение всех мест поездок, визави и контрагентов, наименование товаров и прочее.

5. Программный продукт «Constrim DataRetriever». Программный продукт предназначен для мониторинга открытых источников (Интернет-СМИ, социальных сетей, блогов, форумов, мультимедийных сервисов) и последующего анализа полученных данных. С его помощью можно находить информацию по заданным темам, строить графики количества публикаций, проводить анализ источников, формировать отчеты, экспортировать данные в формате Microsoft Office. Он оперирует такими понятиями, как упоминание и источник. Упоминание – это любая последовательность слов или символов, каким-либо образом связанная с объектом мониторинга. Источником называется ресурс, на котором опубликовано упоминание. Программный продукт «Constrim DataRetriever» обладает следующими возможностями:

- маркировка в ручном режиме тональности упоминаний как позитивную, негативную, нейтральную и неопределенную;
- генерация отчетов системой, в которых отображается тональность упоминаний;
- мониторинг СМИ, представленных в Интернете и доступных для обработки поисковыми машинами;
- мониторинг блогплощадок: Livejournal, LiveInternet, Blogspot, Wordpress, и т.д., Twitter, социальных сетей: Facebook, В Контакте, Flickr, Youtube;

– мониторинг любых других ресурсов по усмотрению пользователя.

– **Система «ABBYU COMPRENO».** Технология Compreno – это универсальная лингвистическая платформа для приложений, решающих множество прикладных задач по обработке текстов на естественном языке. Основной принцип работы – разработка универсальной системы понятий и технологий полного синтаксического и семантического анализа. Второй, но не менее важной частью технологии является полный синтаксический разбор текста. Синтаксис – это способ «кодирования» смысла (семантических отношений) в конкретном языке. Сами семантические отношения универсальны, а способы их реализации в каждом языке – свои. Для каждого языка синтаксическое описание делается заново, но сами средства, которые разные языки используют для кодирования смысла, перечислимы. При описывании нового языка используются разные элементы конструктора (тот же линейный порядок, различные типы синтаксических преобразований, грамматические значения, предлоги, специальные конструкции). Основные возможности системы:

- письменный и устный перевод с одного языка на другой.
- интеллектуальный поиск, в частности: поиск по смыслу, а не по ключевым словам; извлечение фактов и связей между объектами поиска (в т.ч. для конкурентной разведки); мониторинг компаний и персоналий и построение аналитических отчетов на основе параметров разного типа; возможность получить ответы на запросы, заданные обычным языком.
- многоязычный поиск, т.е. когда на вопрос на одном языке находятся ответы на всех языках, поддерживаемых системой.
- классификация и фильтрация документов.
- защита от несанкционированного использования информации.
- автоматическое реферирование и аннотирование документов.
- распознавание речи.

Сравнительный анализ возможностей рассмотренных систем и программных продуктов представлен в табл.1.

Таблица 1

Сравнительный анализ возможностей систем анализа тональности

Система Критерий	Функцио- нальный язык	Работа с мультимедийным контентом	Информация, с которой ра- ботает система	Принцип анализа информации	Требования профессио- нальных навыков	Исполнение системы (ПО)	Генерируемая отчетная документация	Области примене- ния
Аналитический курьер	Кросс-языковая платформа	–	Интернет-страницы, документы	Непрерывный мониторинг	+	Веб-робот	Создание базы данных с результатами	Системы мониторинга
X-Files	Кросс-языковая платформа	+	Локальные досье документов	Непрерывный мониторинг	+	Дистрибутив для ОС Windows	Создание базы данных с результатами	Системы мониторинга и учета
БААЛ	Кросс-языковая платформа	–	Локальные документы	По требованию	–	Интеграция в среду Microsoft Office	Текстовый отчет в формате Microsoft Office	Системы анализа текста
RCO	Русский	–	Локальные документы	По требованию пользователя	–	Набор DLL-библиотек для Windows	Текстовый отчет в формате Microsoft Office	Системы локального поиска
Constrim	Русский, английский	+	Интернет-страницы	По требованию пользователя	+	Дистрибутив для ОС Windows	Текстовый отчет в формате Microsoft Office	Системы мониторинга
ABBY	Кросс-языковая платформа	–	Интернет-страницы, документы	По требованию пользователя	–	Дистрибутив для ОС Windows	Не анонсирована	Системы перевода и системы поиска

В связи с активным развитием и распространением Интернета задача компьютерного анализа тональности текста оказалась востребованной на рынке. Большинство существующих компаний, занимающихся разработкой подобного рода систем, сталкиваются с определенными трудностями:

1. Использование метафор. Пример: «Этот безумный фильм меня впечатлил». В этом предложении присутствует слово «безумный», которое негативно окрашено.

2. Зависимость тональности от контекста. Например, комментарий «прочитай книгу» может быть расценен как позитивный на сайте отзывов о книгах и как негативный на сайте отзывов о фильмах.

3. Обработка тональности зависит от предметной области текста.

4. Привязка системы к одному функциональному языку. Для создания многоязычной системы требуются значительные усилия, сопоставимые с разработкой системы «с нуля».

5. Сложность анализа большого объема текста. В отдельную категорию попадают литературные произведения – книга может быть наполнена большим количеством негативной информации, но выводы, делаемые читателем после прочтения, оставляют о книге положительное мнение. Научить систему анализировать такие тексты подобно человеку – очень сложная задача.

6. Достаточно высокая погрешность анализа. У рассмотренных систем лучшие показатели точности – точность 80% для негативной информации и 60% – для позитивной. В среднем – 60% и 40% соответственно.

7. Сложность распознавания иронии и сарказма.

8. Затрудненный анализ сложных словесных конструкций, предложений, в состав которых входят иностранные слова, устаревшие обороты и т.п.

9. Сложность анализа глубины эмоциональной окраски предложения. Современные системы используют статистические методы подсчета, но они не являются полностью объективными.

10. Методы составления словарей и таблиц тональности слов – трудоемкий ручной способ и сложно реализуемый автоматический.

Несмотря на значительное количество трудностей, которые встречаются на пути разработчиков, данные системы анализа тональности текста являются перспективным направлением для развития IT-технологий. Их реализация позволит создавать новые, более сложные интеллектуальные системы поиска и мониторинга информации, создавать проекты, способные прогнозировать эффект, оказываемый текстом на читателя. Такие

проекты могут быть особенно востребованными в рекламном бизнесе при поиске наиболее удачных названий и торговых марок, помогать при создании учебного материала для школ, ВУЗов и т.д. Системы позволяют осуществлять достаточно эффективный мониторинг социальных сетей, таких как Facebook, Вконтакте, Twitter и др., с целью установления структуры основных тем публикаций, проведения различных тестов и т.д. Применение данных алгоритмов определения тональности текста в различных системах наблюдения и контроля (анализ почтовых клиентов внутри фирмы, факторов динамики развития и роста, анализ клиентов предприятия и т.д.) позволяет пользователю владеть полной и достоверной информацией об окружающей обстановке, предотвращать нежелательные события. Например, в банковском деле – ведение досье на объекты, совершающие подозрительные платежи. В отдельную категорию можно вынести взаимодействие таких систем со специальными службами и правоохранительными органами. Это в первую очередь анализ факторов и динамики развития ситуаций и аналитическая обработка оперативных данных.

В заключении можно отметить, что применение систем анализа тональности текста является достаточно перспективным направлением для развития IT-технологий.

Литература

1. Bollen J., Mao H., and Zeng X.-J. Twitter mood predicts the stock market // Technical Report arXiv:1010.3003, CoRR, 2010, <http://arxiv.org/pdf/1010.3003v1>
2. Пазельская А. Г., Соловьев А. Н. Метод определения эмоций в текстах на русском языке, <http://www.dialog-21.ru/digests/dialog2011/materials/ru/pdf/50.pdf>
3. Гаспаров Б.М. Язык, память, образ. Лингвистика языкового существования. – М.: Новое литературное обозрение, 1996. – 352 с.
4. Осгуд Ч., Дж. Суси, П. Танненбаум. Приложение методики семантического дифференциала к исследованиям по эстетике и смежным проблемам // Искусствометрия. Методы точных наук и семиотики / Под ред.: Лотман Ю. М., Петров В. М.. М.: ЛКИ, 2007г. – С. 278-297.
5. Анализ тональности текста. <http://wikinomica.ru/technology/#>.
6. Киселев С.Л. Системы «Аналитический курьер» и X-Files — основа технологии извлечения знаний текстов из произвольных источников//«Бизнес и безопасность в России" №48/2007, стр. 102-106, <http://www.i-teco.ru/article154.html>
7. Constrim DataRetriever — система мониторинга социальных сетей, блогов и СМИ, <http://www.constrim.ru/cdr>.
8. Реализация и области применения системы ВААЛ, <http://www.vaal.ru/prog/expert.php>.
9. RCO Fact Extractor Desktop – Программа для поиска фактов в тексте, <http://www.rco.ru/product.asp#pd405>
10. Никаноров Г.М., Филиппович А.Ю. Информационная система анализа текстов, <http://it-claim.ru/Education/Seminar/2010/Nikanorov.pdf>
11. Многоцелевой лингвистический процессор ЭТАП-3, <http://www.iitp.ru/ru/science/works/452.htm>

УДК 378.091

АКТУАЛЬНОСТЬ РАЗРАБОТКИ ИНСТРУМЕНТАЛЬНОГО КОМПЛЕКСА ДЛЯ ФОРМИРОВАНИЯ ИНТЕГРИРОВАННОЙ СРЕДЫ УПРАВЛЕНИЯ ОБРАЗОВАТЕЛЬНОЙ ДЕЯТЕЛЬНОСТЬЮ В СИСТЕМЕ СРЕДНЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

Гринченков Д.В., Горяинова О.Н.

Южно-Российский государственный технический университет (НПИ), г. Новочеркасск

Рассмотрены некоторые подходы к формированию автоматизированной системы управления образовательной деятельностью в системе среднего профессионального образования.

В современном мире образование является одним из важнейших факторов, обеспечивающих экономический рост, социальную стабильность, развитие институтов гражданского общества.



Рис. 1. Структура системы образования

Образование, как и всякая социальная подсистема, имеет свою структуру. Система образования включает начальное (НПО), среднее (СПО) и высшее (ВПО) профессиональное образование. Структура системы образования представлена на рис.1 (здесь и в дальнейшем все определения даются на основе действующего закона об образовании).

Начальное профессиональное образование (НПО) – это подготовка работников квалифицированного труда по всем основным направлениям общественно полезной деятельности на базе основного общего образования (9 классов). К современным учреждениям начального профессионального образования относятся:

- профессиональные училища;
- профессиональные лицеи;
- технические школы.

Профессиональные училища осуществляют реализацию образовательных программ начального профессионального образования, обеспечивающих приобретение обучающимися конкретной профессии соответствующего уровня квалификации с получением среднего (полного) общего образования.

Профессиональные училища являются основным типом учреждения начального профессионального образования, в котором осуществляется наибольшая массовая подготовка квалифицированных кадров рабочих и служащих.

Профессиональный лицей – центр непрерывного профессионального образования, осуществляет реализацию интегрированных образовательных программ начального и среднего профессионального образования, обеспечивающих приобретение обучающимися конкретной профессии повышенного уровня квалификации с возможностью получения среднего профессионального образования.

Учебно-курсовой комбинат, учебно-производственный центр, техническая школа, вечернее учреждение профессионального образования осуществляют программу переподготовки, а также подготовки рабочих и специалистов по ускоренной форме обучения.

Среднее профессиональное образование (СПО) – это подготовка специалистов среднего звена на базе основного общего, среднего (полного) общего или начального профессионального образования.

Устанавливаются следующие виды средних специальных учебных заведений:

– *техникум* – среднее специальное учебное заведение, реализующее основные профессиональные образовательные программы среднего профессионального образования базовой подготовки;

– *колледж* – среднее специальное учебное заведение, реализующее основные профессиональные образовательные программы среднего про-

фессионального образования базовой подготовки и программы среднего профессионального образования углубленной подготовки.

Высшее профессиональное образование (ВПО) – это подготовка или переподготовка специалистов на базе среднего (полного) общего, среднего профессионального образования. Виды высших учебных заведений:

1. *Федеральный университет* – высшее учебное заведение, которое:

- реализует инновационные образовательные программы высшего и послевузовского профессионального образования, интегрированные в мировое образовательное пространство;

- обеспечивает системную модернизацию высшего и послевузовского профессионального образования;

- осуществляет подготовку, переподготовку и (или) повышение квалификации кадров на основе применения современных образовательных технологий для комплексного социально-экономического развития региона;

- выполняет фундаментальные и прикладные научные исследования по широкому спектру наук, обеспечивает интеграцию науки, образования и производства, в том числе путем доведения результатов интеллектуальной деятельности до практического применения;

- является ведущим научным и методическим центром.

2. *Университет* – высшее учебное заведение, которое:

- реализует образовательные программы высшего и послевузовского профессионального образования по широкому спектру направлений подготовки (специальностей);

- осуществляет подготовку, переподготовку и (или) повышение квалификации работников высшей квалификации, научных и научно-педагогических работников;

- выполняет фундаментальные и прикладные научные исследования по широкому спектру наук;

- является ведущим научным и методическим центром в областях своей деятельности.

3. *Академия* – высшее учебное заведение, которое:

- реализует образовательные программы высшего и послевузовского профессионального образования;

- осуществляет подготовку, переподготовку и (или) повышение квалификации работников высшей квалификации для определенной области научной и научно-педагогической деятельности;

- выполняет фундаментальные и прикладные научные исследования преимущественно в одной из областей науки или культуры;
- является ведущим научным и методическим центром в области своей деятельности.

4. *Институт* – высшее учебное заведение, которое:

- реализует образовательные программы высшего профессионального образования, а так же, как правило, образовательные программы послевузовского профессионального образования;
- осуществляет подготовку, переподготовку и (или) повышение квалификации работников для определенной области профессиональной деятельности;
- ведет фундаментальные и (или) прикладные научные исследования.

Эффективная работа учебного заведения невозможна без применения в управлении современных информационных технологий. Разработка автоматизированной системы управления учебным процессом позволяет уменьшить временные затраты в процессе планирования и управления деятельностью образовательного учреждения, получить оперативный доступ к информации, сопровождающей учебный процесс, повысить эффективность управления образовательным процессом и учебным заведением в целом.

На современном отечественном рынке автоматизированных информационных систем (АИС) управления учебным представлено достаточно большое количество решений. Проведем сравнительный анализ некоторых наиболее популярных систем:

1. **Конфигурация «1С: Хронограф Школа 3.0 Проф»** – представляет собой новую версию многофункциональной информационной системы электронного документооборота и автоматизации управления основной деятельностью общеобразовательного учреждения.

2. **Программа «1С: Хронограф Школа 3.0 Мастер»** – это система организации и оперативного управления учебным процессом общеобразовательного учреждения.

3. **Программный пакет «Колледж»** предназначен для организации планирования и оперативного управления учебным процессом учреждений начального и среднего образования.

4. **Система Naumen University** – информационно-аналитическая система для организации управления учебным процессом в высших и средних специальных учебных заведениях.

5. Программный продукт «1С: Колледж» представляет собой комплексное решение для управления деятельностью учреждений начального и среднего профессионального образования.

6. Автоматизированная система управления «Колледж» – система, предназначенная для управления учебной деятельностью в системе среднего профессионального образования

7. Автоматизированная информационная система «Университет» представляет собой комплекс программ, объединенных в единую информационную среду и позволяющих автоматизировать процессы, связанные с управлением деятельностью ВУЗа.

8. Комплекс информационных систем «АСУ ВУЗ» предназначен для управления учебным процессом высших учебных заведений.

9. Комплекс программ для автоматизации учебного процесса компании «Аверс» – автоматизированная информационно-аналитическая система управления образовательным учреждением.

Рассмотрена всего лишь небольшая часть систем управления учебным процессом, решающая наиболее важные задачи при автоматизации этого процесса.

Большинство приведенных систем обладают следующими достоинствами:

- модульное построение АИС, обеспечивающее наращиваемость и заменяемость модулей системы;
- единая технологическая база на основе единого ядра в рамках информационного пространства вуза;
- оперативный доступ к единой БД с разделением прав пользователей;
- быстрое внесение изменений в БД;
- интегрированность;
- масштабируемость;
- многофункциональность.

К недостаткам можно отнести то, что обычно АИС управления учебным процессом обеспечивают лишь часть решений, связанных с учебным процессом, но не охватывают все его аспекты. Также было выявлено, что большинство систем в основном решают задачи конкретного учебного заведения с учетом специфики организации учебного процесса в нем. Сравнительный анализ функциональных возможностей этих систем представлены в табл.1.

Таблица 1

Сравнительный анализ систем управления образовательной деятельностью в системе НПО, СПО, ВПО

	НПО		СПО				ВПО		
	Программа «1С:Хронограф Школа 3.0 Проф»	Программа «1С:Хронограф Школа 3.0 Мастер»	Программный пакет «Колледж»	Система Naumen University	Автоматизированная система управления «Колледж»	Программный продукт «1С:Колледж»	Автоматизированная информационная система «Университет»	Комплекс информационных систем «АСУ ВУЗ»	Комплекс программ для автоматизации учебного процесса компаний «Аверс»
Формирование расписания занятий	+	+	+	+	+	+	+	+	+
Учет успеваемости	+	–	–	–	+	+	+	+	–
Учет посещаемости	+	–	–	–	+	+	+	+	–
Управление контингентом студентов	+	–	–	+	+	+	+	+	+
Формирование расписания экзаменов	–	–	–	–	+	–	–	–	–
Автоматизация работы приемной комиссии	–	–	–	+	+	+	+	+	–
Учет аудиторного фонда	+	+	+	+	+	+	+	+	+
Движение персонала	–	+	–	+	–	–	+	+	+
Расчет заработной платы сотрудников	+	–	–	–	–	–	+	+	+
Расчет и выдача стипендии	–	–	–	+	–	–	+	+	+

Среднее профессиональное образование занимает особое место в структуре образования и играет значительную роль в удовлетворении образовательных потребностей населения.

В связи с этим актуальной является задача разработки инструментального комплекса для формирования интегрированной среды управления образовательной деятельностью в системе СПО, с применением качественно нового подхода, который бы позволил систематизировать, формализовать и автоматизировать этот процесс, опираясь на математический аппарат теории массового обслуживания, теории расписаний и теории нечетких множеств, с применением методологии и технологии использования электронных средств обучения.

АИС управления учебным процессом должна решать следующие задачи:

- учет движений контингента студентов;
- распределение нагрузки преподавателей и студентов;
- формирование учебных планов;
- формирования расписания экзаменов;
- составление расписания занятий с учетом внесения в него корректировок и планирования замен;
- анализ результатов успеваемости студентов;
- автоматизация работы приемной кампании.

В рамках создания такой системы должно быть реализовано взаимодействие с ресурсами учебного заведения – финансовым, кадровым и материально-техническим обеспечением.

Следует отметить, что организация управления учебным процессом в системах начального (НПО), среднего (СПО) и высшего (ВПО) профессионального образования имеет некоторые сходства и различия. Рассмотрим более детально основные принципы управления образовательной деятельностью в системе СПО.

1. Организация учебного процесса. Модель организации учебного процесса в системе СПО имеет много общих черт с моделью организации учебного процесса в НПО. Отличием является то, что при формировании расписания должно быть предусмотрено не только деление групп на подгруппы, а также проведение поточных лекций для нескольких групп схожей специализации одновременно.

2. Прием абитуриентов в учебные заведения СПО существенно отличается от приема абитуриентов в учебные заведения НПО. Он осуществ-

ляется на конкурсной основе. Конкурс на прием граждан, имеющих среднее (полное) общее образование, в среднее специальное учебное заведение проводится на основании результатов единого государственного экзамена по общеобразовательным предметам, соответствующим специальности, на которую осуществляется прием. Среднее специальное учебное заведение при приеме по специальностям, требующим наличия у граждан определенных творческих способностей, физических и (или) психологических качеств, вправе проводить по предметам, по которым не проводится единый государственный экзамен, дополнительные вступительные испытания творческой и (или) профессиональной направленности, результаты которых учитываются наряду с результатами единого государственного экзамена при проведении конкурса. В связи с этим в программном комплексе управления образовательной деятельностью в системе СПО должна быть реализована подсистема формирования расписания дополнительных экзаменов.

3. Управление образовательной деятельностью. В средние – профессиональных учебных заведениях промежуточная аттестация осуществляется в форме проведения экзаменов. В реализуемом программном комплексе должна быть создана подсистема формирования расписания экзаменов, которая позволит составить расписание в автоматическом режиме учитывая при этом равномерное распределение экзаменов у групп, занятость аудиторного фонда и нагрузку преподавателей. Также в нем должен проводиться интеллектуальный анализ успеваемости с возможностью выделения критических точек реализации образовательного процесса и автоматической выработкой рекомендаций по разрешению проблем.

Рассмотрев управление образовательным процессом для всех уровней образования, можно сделать вывод, что организация этого процесса в системе СПО имеет существенные отличия с НПО и очень схожа с ВПО, однако главным отличием является масштабность высших учебных заведений. Это отличие позволяет накладывать более строгие ограничения при разработке систем СПО, связанных с учебным процессом, что обеспечивает получение результата, наиболее близкого к оптимальному, в автоматическом режиме.

Функциональная структура программного комплекса для управления учебным процессом в системе СПО представлена на рис.2.

На рис. 3 и 4 представлена детализация блоков «Организация образовательного процесса» и «Управление контингентом учащихся».



Рис. 2. Функциональная структура системы



Рис. 3. Организация образовательного процесса



Рис. 4. Управление контингентом учащихся

УДК 378.091

БАЗОВЫЕ ПРИНЦИПЫ РАЗРАБОТКИ И ПРИМЕР РЕАЛИЗАЦИИ ПОИСКОВОГО РОБОТА

Гринченков Д.В., Куций Д.Н.

Южно-Российский государственный технический университет (НПИ), г. Новочеркасск

Рассмотрены базовые принципы и подходы, реализуемые сегодня при разработке поисковых роботов. Приводится пример реализации поискового робота выполненный на языке Java, а так же демонстрируются результаты его тестирования, подтверждающие работоспособность выбранного подхода.

Поисковый робот представляет собой встроенный в поисковую систему программный модуль, подчиняющийся ее алгоритмам. Робот анализирует содержание опрашиваемой страницы на соответствие ключевых слов и сохраняет результат. Получаемые в результате обхода сведения являются базой для дальнейшей работы поисковой системы путем их последующей обработки индексатором.

В зависимости от назначения робота можно выделить следующие подходы к опрашиванию страниц: опрос по конкретному списку и самостоятельное построение списка опрашиваемых страниц.

Первый подход применим в том случае, если есть фиксированный список страниц, за которыми нужно наблюдать и отслеживать их обновления. Однако полученная таким способом база данных может не отвечать условия полноты поисковой системы. Решить данную проблему можно следующим способом: поисковый робот в качестве исходных данных получает список не конкретных страниц для опроса, а список «точек входа». Опросив такие страницы, он получает список ссылок от «точек входа» и на следующем этапе опрашивает уже их. Этот процесс продолжается до тех пор, пока не сработает условие прекращения работы алгоритма.

К критериям остановки поиска можно отнести объем анализируемого текста и проверка в глубину или в ширину. При поиске в глубину робот опрашивает точку входа и совершает еще некоторое число переходов по ссылкам в глубину, количество которых не превышает некоторого заранее заданного значения. При поиске в ширину проверяется каждая ссылка на странице, прежде чем перейти к следующей странице.

При использовании этих критериев можно обработать ту часть ссылок, по которым перейдет пользователь от некоторой начальной точки, а сам робот прекратит свою работу в результате срабатывания критерия остановки.

Помимо этого необходимо учесть факт ограничения доступа поискового робота к определенным страницам web-сайта, установленные в файле robot.txt. Таким образом, робот-поисковик должен хранить набор ссылок, по которым нельзя переходить.

Также необходимо учитывать, что страницы могут ссылаться друг на друга, что приводит к необходимости проверки дублирования найденных ссылок.

С учетом представленной информации общий алгоритм работы поискового робота представлен на рис. 1.

Для создаваемого поискового робота использован следующий набор настраиваемых параметров:

1. Список «точек входа».
2. Ширина обхода.
3. Время ожидания от сервера (по истечению этого времени страница помечается как несуществующая).
4. Регулярные выражения для поиска ссылок на других страницах.

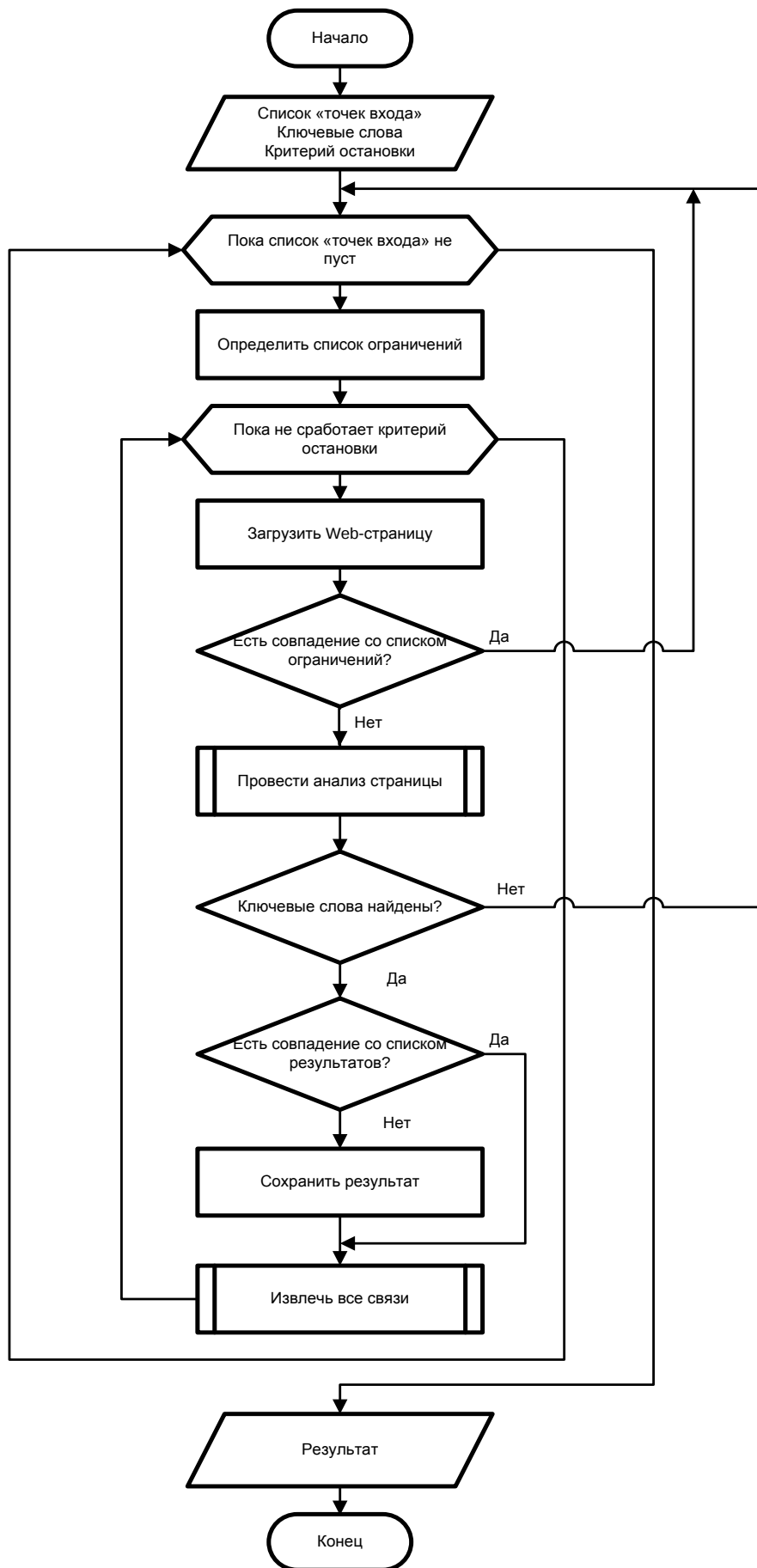


Рис. 1. Общий алгоритм работы поискового робота

Эффективность проверки прав доступа к некоторому заданному URL обеспечивается размещением списка ограничений в кэш-памяти сразу после считывания сайта. Просмотр кэш-памяти устраняет необходимость циклической загрузки списка ограничений для проверки каждого URL, что приводит к повышению производительности за счет экономии времени.

Список точек входа обрабатывается последовательно, пока не будут опрошены все страницы, связанные с текущей начальной точкой, другие элементы исходного списка не рассматриваются. Поскольку процедура обхода страниц достаточно ресурсоемкая, для запуска процесса поиска создается новый поток с целью распределения выполнения основного исполняемого модуля и процесса обхода страниц. Такое решение позволяет не только решить проблему переполнения стека рекурсивными вызовами, но и ускорить сам процесс обхода страниц при загрузке.

Для обнаружения ссылок на странице используются регулярные выражения API со следующим шаблоном:

$(?<=<a\ [^<]*href\ *=\ *(")|)[^#"]\.*?(?=(")|>?)$, где

$(?<=<a\ [^<]*href\ *=\ *(")|)[^#"]\.*?(?=(")|>?)$ – Позитивный просмотр назад. Реализует способ производить поиск фрагмента текста, не включая в найденный результат окружающий текст, который расположен до или после искомого фрагмента текста. С его помощью находим имя тега HTML, не включая в результат поиска окружающие его угловые скобки или другие знаки, но и не упуская их при поиске нужного контекста.

$<a$ – Поиск последовательности " $<a$ ".

$[^<]*$ – Поиск ноль или более символов, кроме $<$.

$href\ *$ – Поиск последовательности символов "href" ноль или более раз.

$=\ *$ – Поиск последовательности символов '=' ноль или более раз.

$("|")$ – Поиск любого из указанных символов.

$[^#"]$ – Поиск любого символа, кроме '#' и символа кавычек.

$\.*?$ – Поиск одного или нескольких любых символов, пока не встретится следующая часть шаблона.

$(?=шаблон)$ – Позитивный просмотр вперед.

$("|")>?$ – поиск последовательности " $>$ " или " $'>$ " ноль или более раз.

Примеры соответствия шаблону: $<a\ href="http://abeent.co.cc">a$,

$<a\ href="http://\ www.twitterbackgrounds.org/"\ target="_blank">$.

Строка, содержащая ключевые слова для поиска, разбивается на отдельные термы, после чего производится циклический просмотр отдельных термов для определения совпадений с содержимым страницы с помощью алгоритма Бойера-Мура. Поиск не чувствителен к регистру.

Поисковый робот реализован на языке Java, поскольку стандартные средства языка позволяют работать с web-страницами, а имеющиеся библиотеки упрощают реализацию фрагмента поиска связей и сохранения списка связей.

Работу поискового робота можно продемонстрировать на следующем тестовом примере. Для начала в списке URL можно задать один адрес для общего отслеживания процесса (рис. 2).

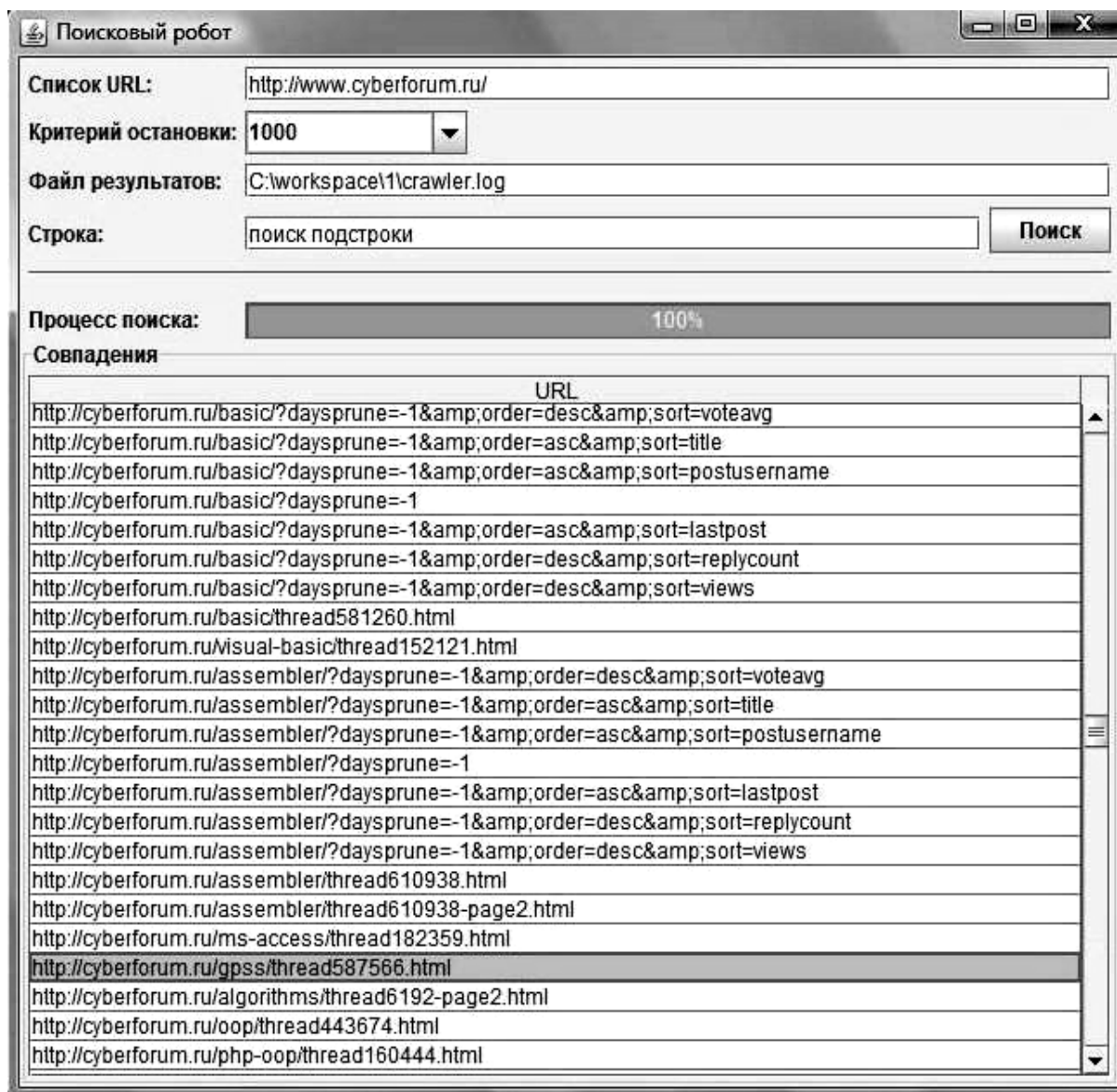


Рис. 2. Тестирование поискового робота

Для проверки полученных результатов выберем любую отображенную в списке результатов страницу, загрузим ее и проверим наличие ключевых слов в тексте страницы (рис. 3).

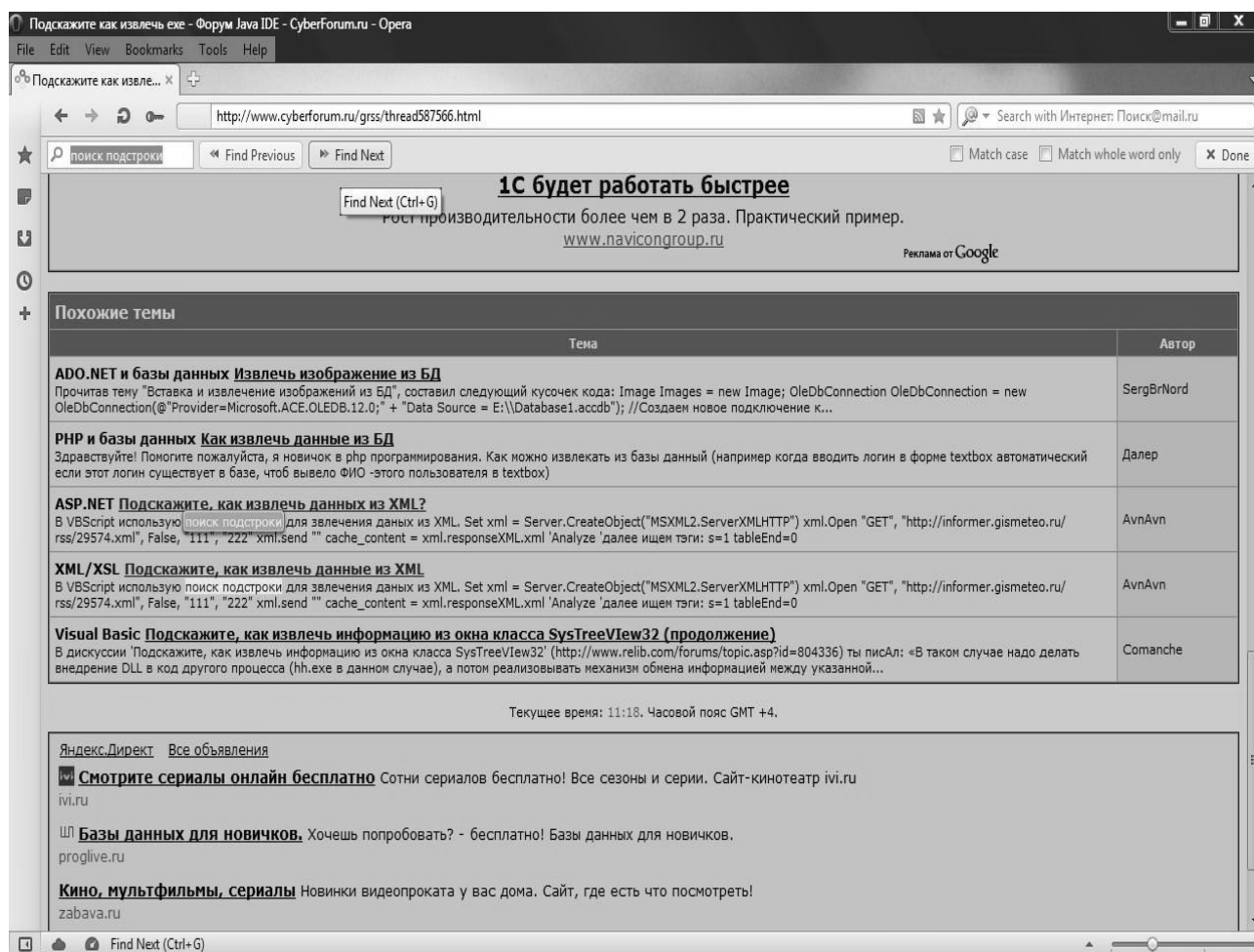


Рис. 3. Проверка результатов

Однако ранее оговаривалась возможность работы со списком адресов. В качестве стартовых точек зададим адреса двух форумов по программированию для поиска страниц, на которых находятся ключевые слова (<http://www.cyberforum.ru> и <http://programmersforum.ru>). На рис. 4 видно, что список точек входа обрабатывается последовательно. Число адресов ограничено, поскольку на данном примере иллюстрируется только функциональная возможность работы с несколькими стартовыми точками с последующими переходами вглубь указанных сайтов.

Функциональность данного модуля позволяет решать базовые задачи работы поискового робота: выполнять обход страниц и сохранять результаты рабочей сессии, а обладает достаточной гибкостью для дальнейшей модификации с целью дальнейшей разработки поискового робота для конкретных задач тематического поиска.

Все результаты обхода страниц по ссылкам сохраняются в одном файле. Формат сохраняемых результатов должен быть разработан

так, чтобы в дальнейшем с ним могли работать остальные модули поисковой системы.

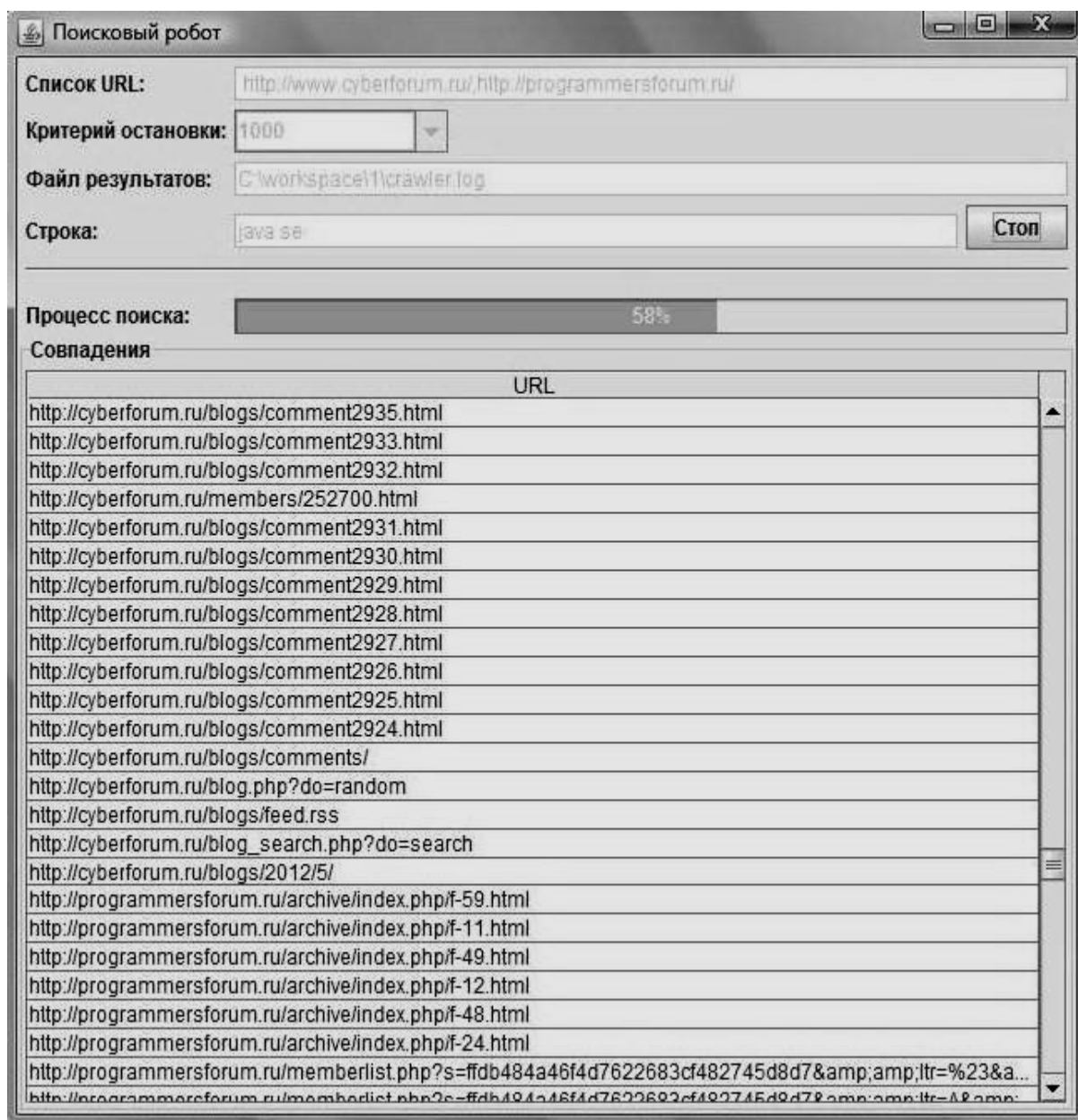


Рис. 4. Тестирование поискового робота для списка адресов

Литература

1. Опыт написания поисковой системы <http://php.su/articles/?cat=examples>
2. http://ru.wikipedia.org/wiki/Алгоритм_Бойера_–_Мура
3. Поиск в строках, массивах, последовательностях <http://algolist.manual.ru/search/>
4. Простой поисковый робот <http://juravskiy.ru/?p=1005>
5. <http://www.seonews.ru/masterclasses/detail/29814.php>
6. <http://blog-raskruti.ru/blogosfera/poiskovyiy-robot-robotyi-poiskovyih-sistem>
7. Регулярные выражения <http://www.rsdn.ru/article/alg/regular.xml>

УДК 378.091

ПРИНЦИПЫ ПОСТРОЕНИЯ ИНТЕЛЛЕКТУАЛЬНОЙ СИСТЕМЫ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ WEB-ПРИЛОЖЕНИЙ

Гринченков Д.В., Шаменко А.А.

Южно-Российский государственный технический университет (НПИ), г. Новочеркасск

Рассматриваются принципы построения интеллектуальной системы автоматизированного проектирования web-приложений. Описаны подходы к формированию эффективного инструментария позволяющего управлять процессом проектирования приложения, обеспечивающего разработчику экспертную поддержку при принятии решений, минимизирующего необходимый уровень специализированных знаний в области программирования.

В настоящее время при разработке web-приложений используется множество различных технологий и языковых средств. Основными из них, т.е. используемыми в большинстве проектов, можно считать следующие: HTML, CSS, JavaScript, PHP и MySQL. Каждая из них предназначена для решения определенных задач. Взаимодействие между перечисленными технологиями представлено на рис.1.

В этой ситуации задачи автоматизации можно определить следующим образом:

1. Организация поточного производства web-приложений на основе готовых шаблонов. Небольшие компании обычно предпочитают создание сайта на основе шаблона. При этом существующий образец (шаблон) видоизменяют в зависимости от пожеланий заказчиков. Это позволяет ускорить производство, но требует хорошее знание самого шаблона, т.е. того как он работает и какие ресурсы при этом используются. Наличие комментариев может частично решить эту проблему, но не в случае изменения шаблонов с большим количеством страниц с различным оформлением. Таким образом, ограничение использования шаблонов может быть связано с масштабом создаваемых web-приложений. И оно может быть снято, если при изменении шаблона не будет необходимости вникать в структуру страницы и в то, как она устроена. Экспертная система позволяет решить эту проблему именно так. Опрос пользователя экспертной системой играет роль комментариев, при этом данные опроса интерпретируются системой для изменения шаблона. Так как система сама управляется с кодом, то пользователю нет необходимости, что-то делать самому.

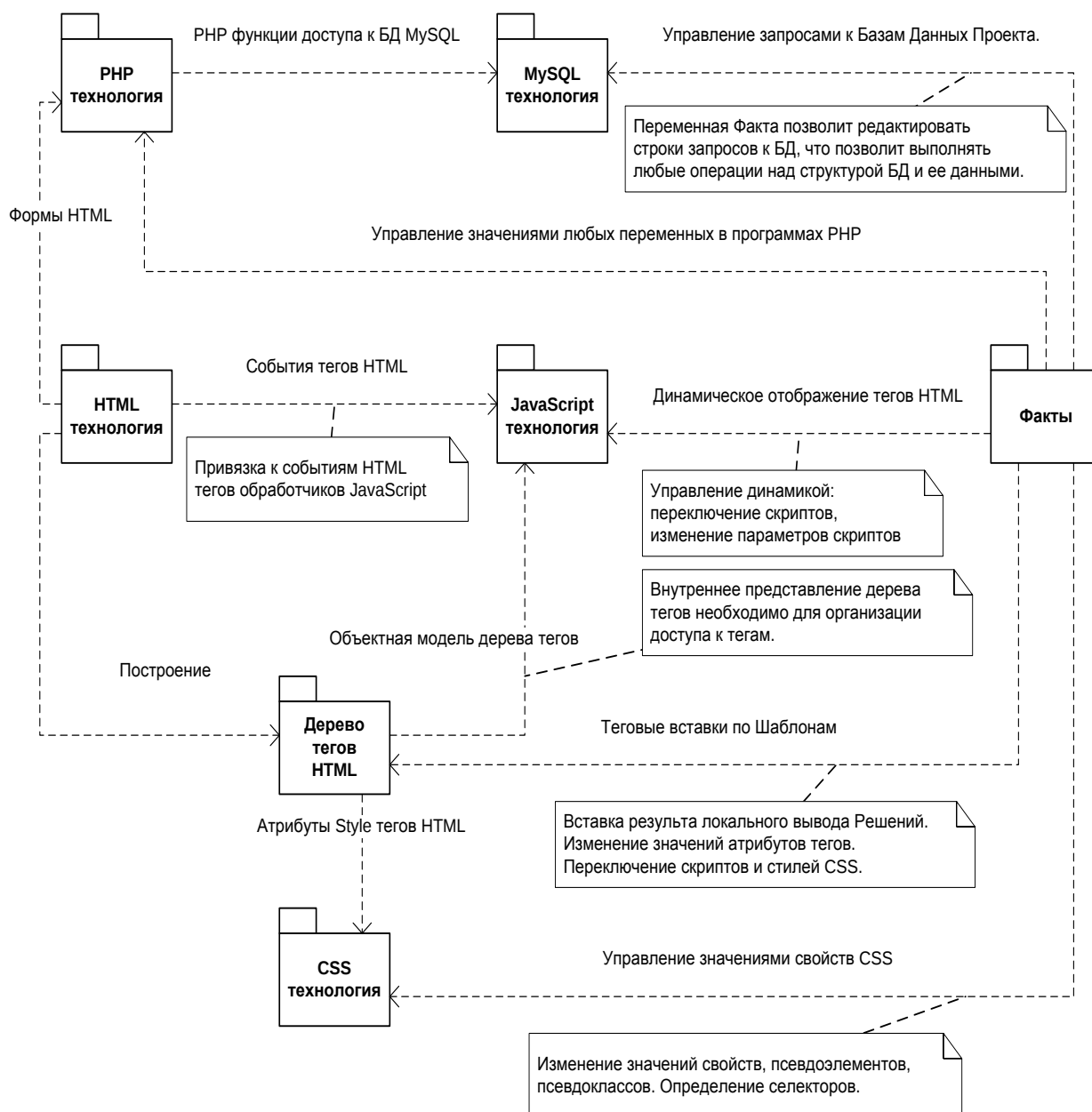


Рис. 1. Взаимодействие между технологиями разработки web-приложений

В результате разработка web-приложения может быть сведена к прохождению опроса, на основе которого система вывода сама сможет принять решение о том, какой шаблон использовать, как компоновать страницы, какие технологии следует использовать. Задача разработчика-дизайнера при этом заключается в том, чтобы правильно задать направление разработки, т.е. выбрать правильный шаблон и далее заполнить данными страницы и другие хранилища. Данная схема позволит ускорить производство web-приложений, так как стадия изменения шаблонов перейдет в стадию оформления страниц дизайнером. Сосредоточив внимание

на оформлении и дизайне, можно существенно повысить качество конечного результата или, наоборот, за счет него ускорить производство.

2. Абстрагирование дизайна от разработки. Технологии, используемые при создании web-приложений, могут быть достаточно точно структурированы, а значит ими в некоторой степени можно управлять. Например, HTML в своем составе имеет теги, атрибуты тегов и значения которые принимают эти атрибуты и на этом многообразии сущностей HTML заканчивается. Управление кодом может осуществляться при помощи знаний экспертной системы: фактов и правил. Создавая модели управляемых программ и выделяя участки управляемого кода, система вывода сможет корректировать их при помощи опроса пользователя, т.е. на основе вводимых пользователем данных. В этом свете экспертная система предстаёт как связующее звено между пользователем и кодом, созданным программистом и его опытом, заложенным в базу знаний. Абстрагирование создается на основе знаний программистов и разработчиков шаблонов, а реализуется за счет экспертной системы изменяющей шаблоны на основе этих знаний. В результате кодирование шаблона может быть полностью отделено от его дальнейшего использования.

Решение поставленных задач автоматизации позволяет ускорить производство web-приложений за счет использования шаблонов и разделения труда. Значительную роль при этом имеет применение экспертной системы, как связующего звена между категорией пользователей эксперты, то есть специалистов по различным технологиям и категорией пользователей дизайнеры.

Фундаментом экспертной системы любого типа является база знаний, которая составляется на основе экспертных знаний специалистов. В таком представлении база знаний экспертной системы станет основой для работы производителей web-приложений, что в результате даст независимость от текучести кадров и обезопасит разработчиков от потерь связанных с уходом специалистов. Опыт разработок и знания специалистов будут накапливаться в базе знаний.

Пополнение базы знаний (базы шаблонов) может выполняться в фоновом режиме независимо от наличия разрабатываемых проектов у производителей web-приложений, а при наплыве заказов может быть полностью остановлено и возобновляться только при необходимости в новых шаблонах. Интеллектуальная система, связанная с базами данных технологий, которые должны содержать подробное описание и данные о структуре технологий, а также синтаксисе, валидации (HTML и CSS), позволит

корректировать шаблоны непосредственно при их использовании дизайнером, а не при описании их экспертом.

Одной из основных проблем, возникших при проектировании, стала управляемость кодом, которая заключалась в создании системы способной управлять исходным кодом различных технологий с различными правилами синтаксиса и принципами применения. При ее решении можно выделить следующие подзадачи.

Во-первых, представление знаний. Все технологии, используемые при разработке web-приложений, имеют различный синтаксис и принцип использования. Таким образом, возникает проблема их унификации, то есть создания единого представления синтаксиса этих технологий. Подвести все технологии к единому знаменателю невозможно – слишком большое количество различных конструкций. Представление конструкций всех технологий должно быть адаптировано под их использование, то есть разрабатываться с учетом их особенностей. Решением является создание собственной системы представления кода различных языков. Сам код при этом рассматривается как знания, хранимые в экспертной системе.

Для решения этой подзадачи эффективным решением может быть использования языка XML. Применение XML позволяет создать собственный декларативный язык представления знаний, который можно рассматривать как встроенный язык системы автоматизированного проектирования web-приложений. Для каждой рассмотренной технологии возможно создание собственного представления языковых конструкций и синтаксиса. Пример модели представления знаний технологии HTML представлен на рис. 2.

Для представления знаний всех уровней используется понятие шаблона. Шаблоны можно рассматривать как контейнеры или пространства имен связующие вместе множество фактов и правил относящиеся к некоторому подмножеству знаний из общей базы. Данные, к которым применяются факты и правила, также находятся в шаблоне и ими нельзя управлять из других шаблонов, хотя разрешается взаимодействовать с глобальным пространством объектов приложения. Уровни знаний и используемые шаблоны представлены на рис. 3.

Связи между моделями знаний различных технологий повторяют реальную связь между этими технологиями. Диаграмма связи между моделями представления исходного кода технологий и моделями управляющих конструкций экспертной системы представлена на рис. 4.

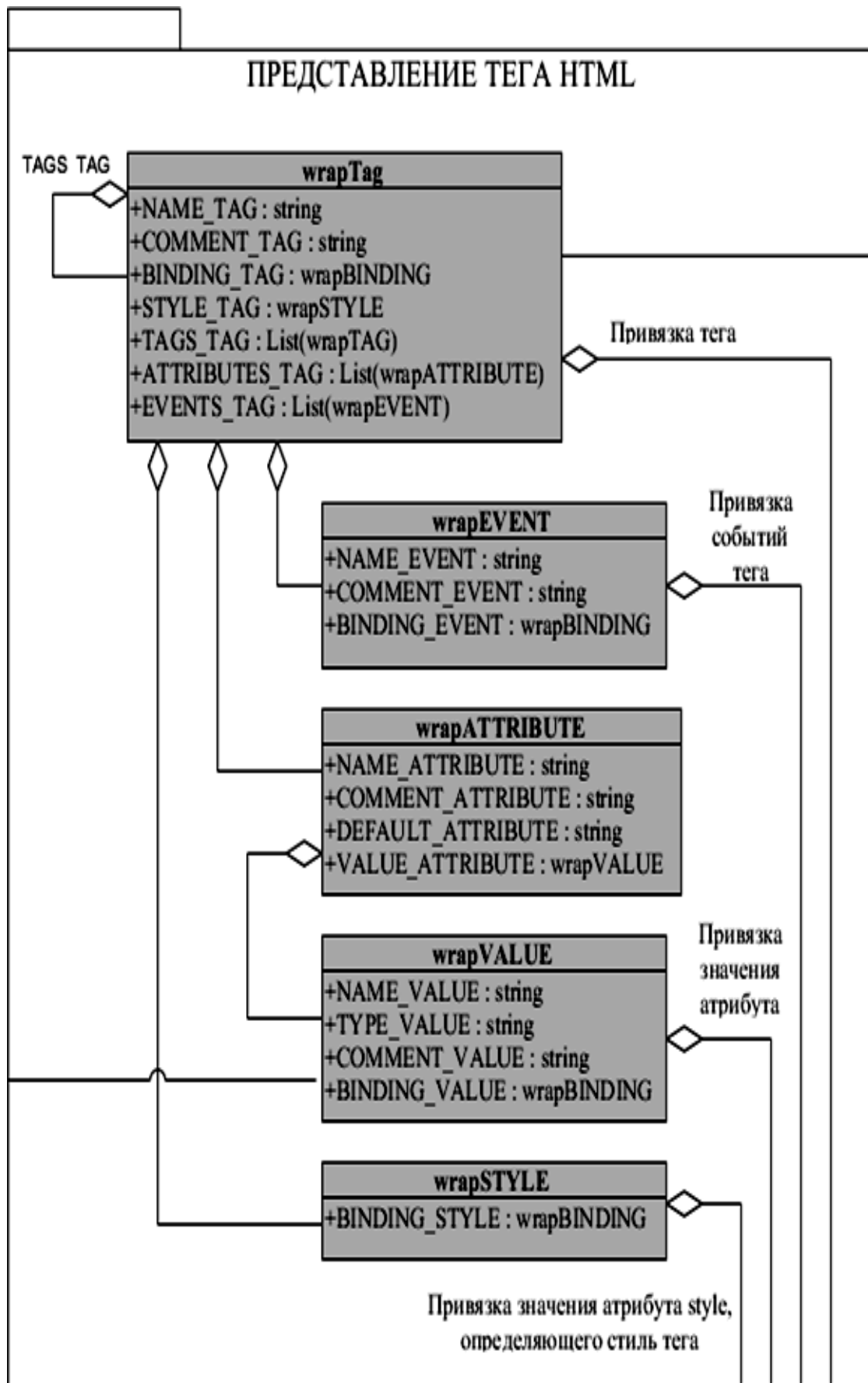


Рис. 2. Модель представления тега HTML

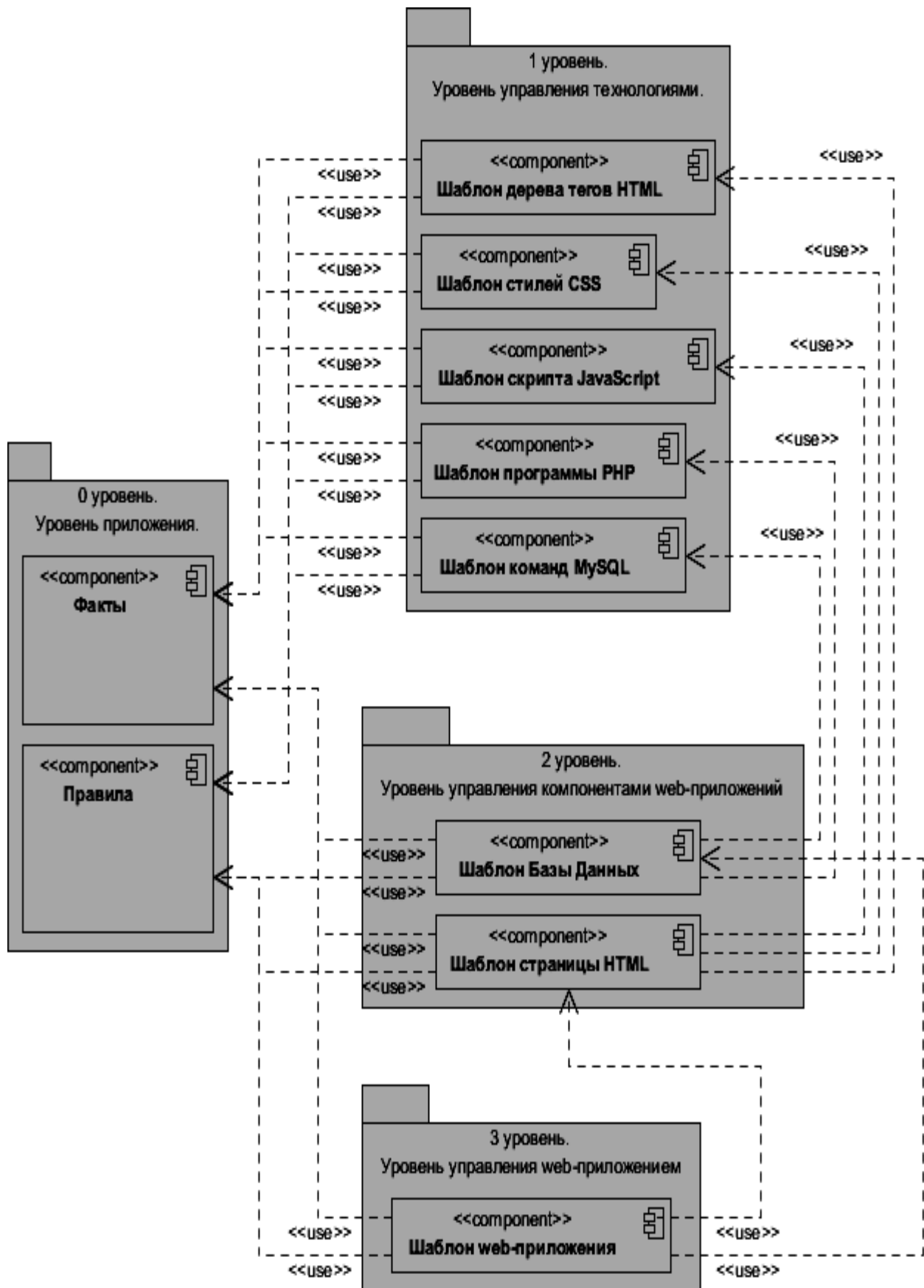


Рис. 3. Уровни знаний и используемые шаблоны

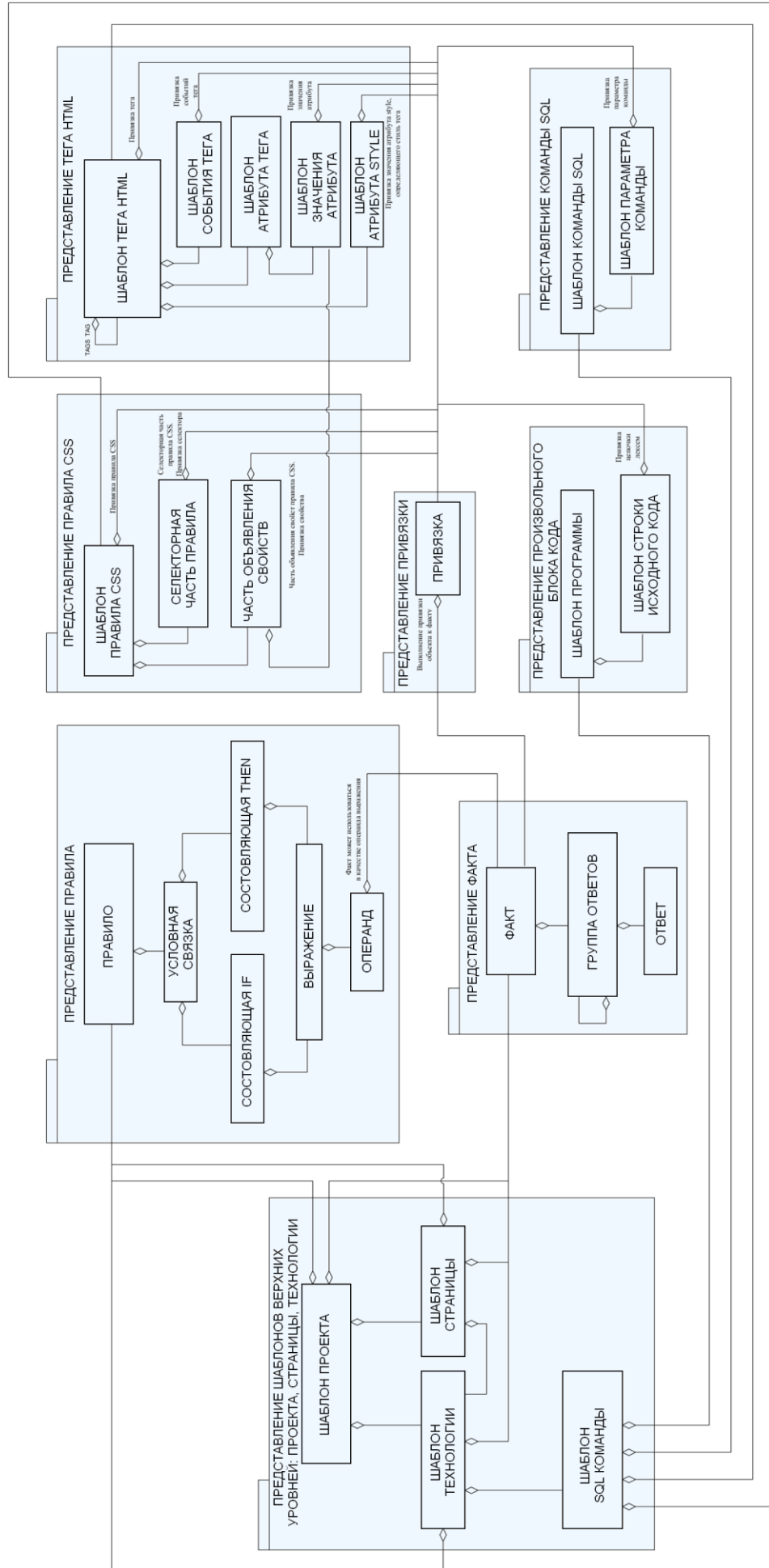


Рисунок 4 – Диаграмма представления данных

Во-вторых, способы управления кодом. Были выделены следующие способы управления кодом:

1. Использование конструктора исходного кода. Этот способ заключается в построении модели кода при помощи формы-мастера. На форме располагаются кнопки для ввода элементов различных конструкций в модель, также отображается сама модель в виде списков или дерева. Результат работы мастера – полноценную модель исходной программы в памяти компьютера. Аналогом данного метода является конструктор SQL запросов, используемый в Visual Studio. Этот метод может оказаться очень трудоемким для пользователя в случае составления программ на языках с большим разнообразием синтаксических конструкций. Для декларативных языков, таких как HTML, он подходит идеально, работа пользователя при этом сводится к построению дерева тегов, а база конструкций HTML предоставит для этого всю необходимую информацию.

2. Использование лексического анализа с целью выявления языковых конструкций, которыми можно управлять. Этот способ основан на использовании лексического анализатора, т.е. выделении в нем конструкций (лексем) к которым может быть выполнена привязка управляющих конструкций экспертной системы. Главная задача лексического анализатора состоит именно в поиске управляемых конструкций. Так синтаксический анализ, выполняемый на основе найденных лексем, фактически является дополнительной функцией системы, например, для подсветки кода или проверки его правильности. Можно выделить две группы лексем: лексемы управления и лексемы для синтаксического анализа.

Структурный анализ технологий позволяет принять решение о том, какой способ управления кодом должен быть выбран. Так HTML – простой, достаточно структурированный язык, поэтому для управления деревьями тегов может быть использован конструктор кода, с использованием базы данных языковых конструкций. Такой подход позволит управлять всеми элементами кода HTML. На рис. 5 представлена диаграмма способов управления кодом.

В-третьих, проблема связывания шаблонов – моделей представления кода технологий с возможностями экспертной системы, которые ограничены использованием управляющих структур: фактов и правил. Решением данной проблемы стало внедрение понятия привязки. Привязка – ссылка на объект, который определяет некоторое значение. Привязка позволит заменить лексему в коде на значение, определяемое экспертной системой. На основе этого понятия был создан конструктор привязок, выполняющий связывание шаблонов различных уровней и типов при помощи привязок.

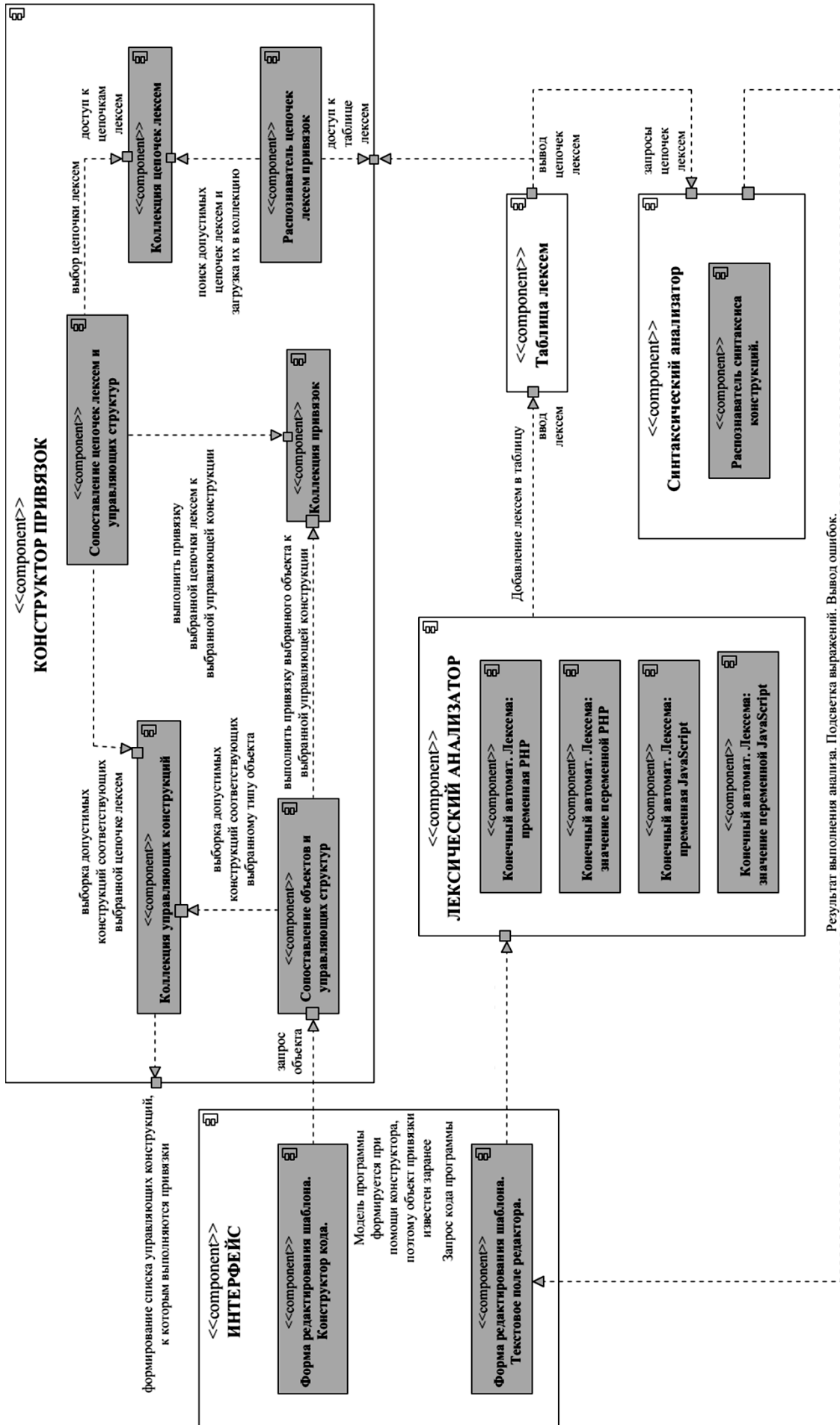


Рисунок 5 – Диаграмма способов управления кодом

Привязки различаются по категориям в зависимости от используемого шаблона. На рис.6 показана структурная схема конструктора привязок выполняющего привязку цепочек лексем к управляющим структурам. Данная схема применяется совместно с лексическим анализом. При использовании конструктора кода сопоставление управляющих конструкций выполняется к построенной пользователем модели исходного кода.

Вторая стадия проектирования полностью заключалась в разработке экспертной системы. На рис. 7 представлена структурная диаграмма экспертной системы.

Экспертная система позволяет решить поставленные задачи автоматизации: абстрагирования и организации поточного производства. За счет: использования моделей представления знаний для создания шаблонов; управляющих конструкций; привязок к управляющим конструкциям; и системы вывода, которая управляет шаблонами на основе выполняемых с пользователем консультаций.

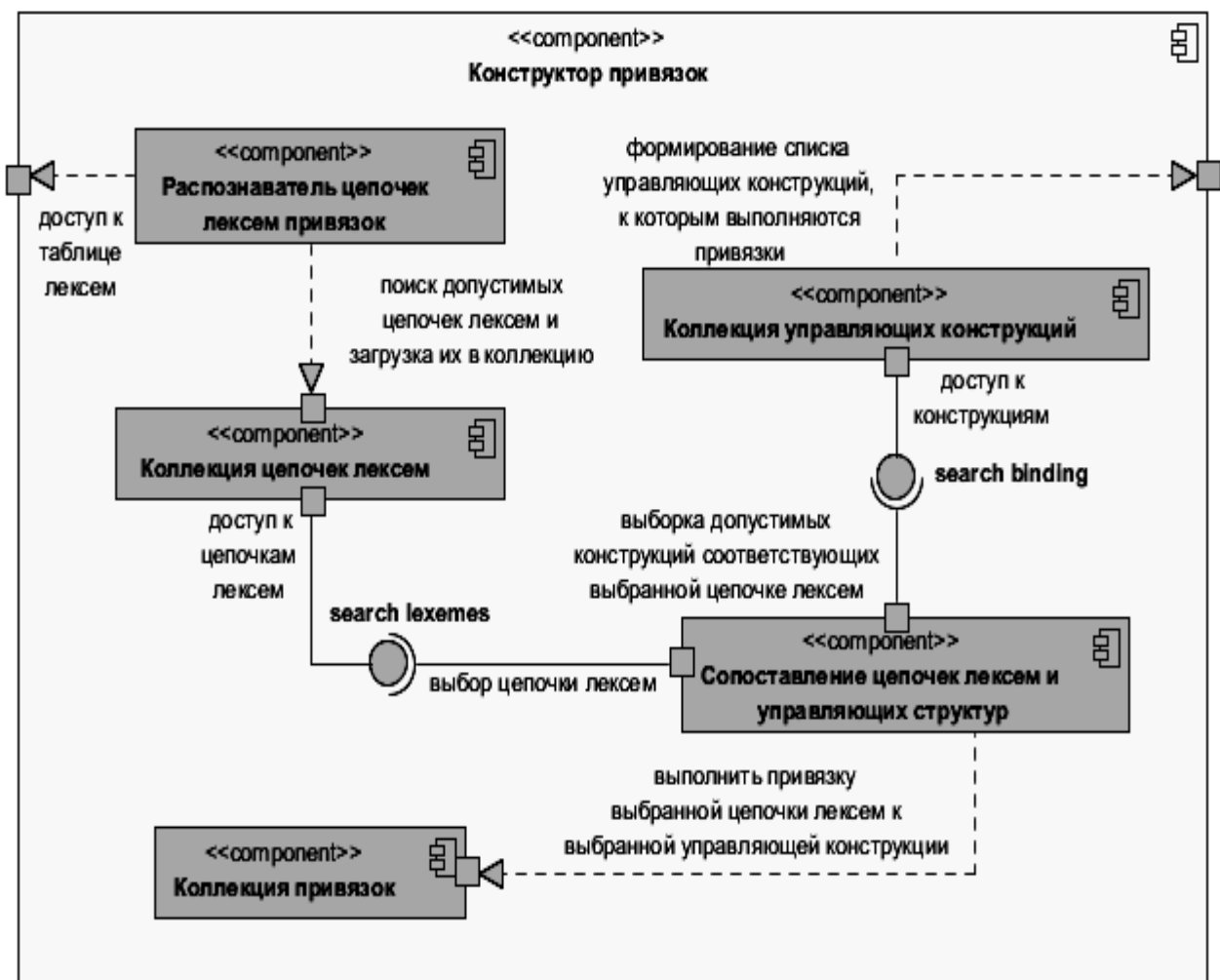


Рис. 6. Конструктор привязок

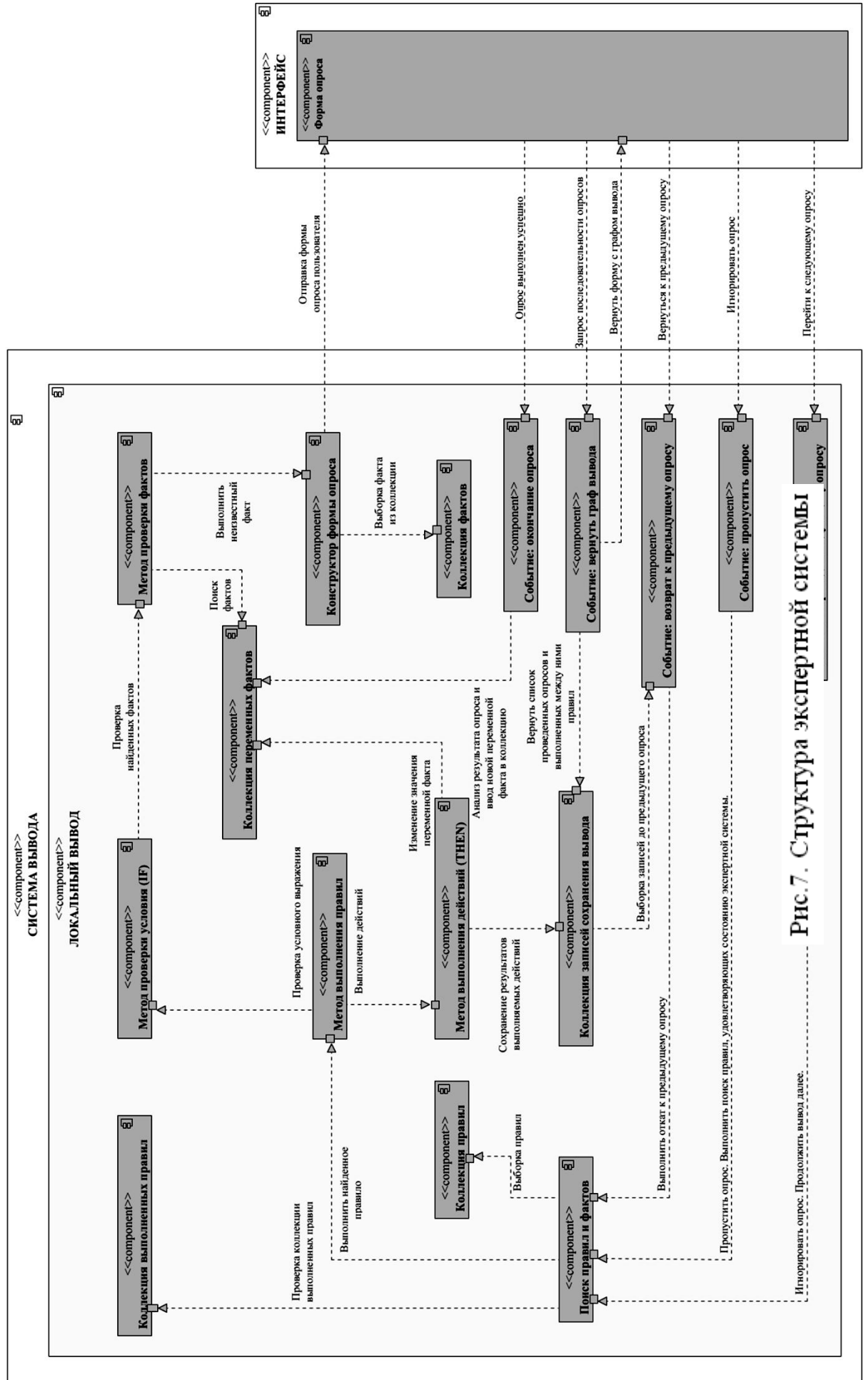


Рис 7. Структура экспертной системы

Построение экспертной системы выполнялось на основе событийной архитектуры, так при выполнении любых действий экспертная система обрабатывает события интерфейса пользователя. Для взаимодействия с пользователем экспертной системой динамически создаются формы опроса. В подобных формах может быть отображена вспомогательная информация: графы опросов, пояснения экспертной системы или комментарии эксперта.

Система сохранения последовательностей выводов также управляемая экспертной системой позволяет создавать проекты и дает возможность быстро изменить последовательность вывода при необходимости. Система сохранения вывода позволяет модернизировать создаваемые web-приложения и обеспечивает их сопровождаемость.

УДК 004.051

ЭФФЕКТИВНОСТЬ ИНФОРМАЦИОННЫХ СИСТЕМ УПРАВЛЕНИЯ

Дущенко Т.К., Нечаев А.В.

Новочеркасская государственная мелиоративная академия

Одной из важнейших экономических проблем является оценка эффективности использования информационных систем в управлении предприятиями.

Хозяйственная деятельность организаций во всех сферах экономики неразрывно связана с получением, накоплением, хранением и использованием различных сведений, характеризующих как само предприятие, так и связанных с ним хозяйствующих партнеров и конкурентов. Информация сегодня рассматривается в качестве одного из важнейших ресурсов развития общества наряду с материальными, энергетическими и человеческими ресурсами. Более того, эффективность производственно-хозяйственной деятельности в современных условиях находится в прямой зависимости от эффективности использования информационных ресурсов.

В соответствии с Федеральным законом РФ «Об информации, информатизации и защите информации» **информационные ресурсы** – это отдельные документы или массивы документов в информационных системах. **Информационная система (ИС)** – организационно упорядоченная совокупность документов (массивов документов) и информационных технологий, в том числе с использованием средств вычислительной техники и

связи, реализующих **информационные процессы** – процессы сбора, обработки, накопления, хранения, поиска и распространения информации. Главной составной частью информационной системы является **информационная технология (ИТ)** призванная обеспечивать эффективное использование информационных ресурсов.

По мнению автора [7] с помощью информационных технологий достигают повышения скорости и качества обработки информации, а также колоссального увеличения объемов хранимой и обрабатываемой информации, что сказывается в конечном итоге на эффективности управления организацией. В данной статье рассматриваются организации, для которых информационные технологии являются средством совершенствования управленческой деятельности, способствуют основной деятельности, но не участвуют в ней непосредственно. В этом случае ИТ содействуют более качественному управлению организацией, удешевляют и повышают эффективность принимаемых управленческих решений и таким опосредованным образом способствуют повышению конкурентоспособности [5].

Эффективность как одно из наиболее общих экономических понятий, не имеет на текущий момент единого общепризнанного определения. В ходе активно ведущихся дискуссий приводится множество различных определений этого понятия. По одному из мнений [6], **эффективность** – это одна из возможных характеристик качества некоторой системы (экономической, технической, социальной, экологической, информационной), а именно ее характеристика с точки зрения соотношения затрат и результатов функционирования системы.

В нормативных документах понятие эффективности закреплено ГОСТ Р ИСО 9000 - 2001 «Системы менеджмента качества. Основные положения и словарь»: эффективность – соотношение между достигнутым результатом и использованными ресурсами. Определение эффективности приводится Методическими рекомендациями по оценке эффективности инвестиционных проектов № ВК 477 от 21 июня 1999 года. **Эффективность инвестиционного проекта (ИП)** – категория, отражающая соответствие проекта, порождающего этот ИП, целям и интересам участников проекта. В этом документе подчеркивается, что термин «**экономическая эффективность**» используется в более узком смысле – как обозначение одной из характеристик эффективности, а именно эффективности инвестиционного проекта с точки зрения экономики общества в целом.

Представляется также целесообразным определить с понятием виды эффективности. Автором [7] приводится перечень видов эффектив-

ности и допускается подмена понятий. Обратимся к методическим рекомендациям, в которых говорится, что «осуществление эффективных проектов увеличивает поступающий в распоряжение общества ВВП, который затем делится между участвующими в проекте субъектами (фирмами (акционерами и работниками), банками, бюджетами разных уровней). Поступлениями и затратами этих субъектов определяются различные виды эффективности ИП». Рекомендуются оценивать следующие виды эффективности:

- 1) эффективность проекта в целом;
- 2) эффективность участия в проекте.

Эффективность проекта в целом включает в себя:

- 1) общественную (социально-экономическую) эффективность проекта;
- 2) коммерческую эффективность проекта.

Эффективность участия в проекте включает:

- 1) эффективность участия предприятий в проекте;
- 2) эффективность инвестирования в акции предприятия;
- 3) эффективность участия в проекте структур более высокого уровня по отношению к предприятиям, в том числе:

3.1) региональную и народнохозяйственную эффективность — для отдельных регионов и народного хозяйства РФ;

3.2) отраслевую эффективность — для отдельных отраслей народного хозяйства, финансово-промышленных групп, объединений предприятий и холдинговых структур;

4) бюджетную эффективность, то есть эффективность участия государства, муниципалитета в проекте с точки зрения расходов и доходов бюджетов всех уровней.

Система показателей, определяемая для оценки перечисленных видов эффективности, и методологические принципы их расчета едины. Отличия заключаются в тех исходных параметрах, которые формируют потоки реальных денежных средств по проекту применительно к каждому виду эффективности. В качестве параметров учитываются денежные притоки и оттоки от операционной, инвестиционной и финансовой деятельности.

Методикой расчета показателей и применения критериев эффективности инвестиционных проектов, претендующих на получение государственной поддержки за счет средств Инвестиционного фонда РФ № 139/82н от 23 мая 2006 года, выделяются качественные и количественные критерии отбора инвестиционных проектов. Количественные критерии позволяют оценить финансовую, бюджетную и экономическую эффективность ИП.

Финансовая эффективность проекта учитывает финансовые последствия реализации проекта для ее непосредственных участников. Оценка финансовой эффективности осуществляется на основе построения финансовой модели инвестиционного проекта, которая включает связанные прогнозные отчетные формы – отчет о прибылях и убытках, балансовый отчет и отчет о движении денежных средств, выполненные на срок прямого прогнозирования денежных потоков. Показателями финансовой эффективности являются: чистая приведенная стоимость (Net Present Value, NPV), внутренняя норма доходности (Internal Rate of Return, IRR), период окупаемости инвестиционного проекта (Payback Period, PP) и удельная финансовая эффективность инвестиционного проекта (RFA). В качестве показателя **бюджетной эффективности** используется индекс бюджетной эффективности (PI).

Экономическая эффективность инвестиционного проекта оценивается по его способности влиять на формирование внутреннего валового продукта (ВВП) экономики и обеспечивать динамику экономического роста. Экономическая эффективность отражает воздействие реализации инвестиционного проекта на внешнюю для проекта среду и учитывает соотношение результатов и затрат по проекту, которые прямо не связаны с финансовыми интересами участников проекта и могут быть количественно измерены.

Применительно к оценке эффективности внедрения ИТ на предприятии следует использовать термин «эффективность». Эффективность ИТ – это комплексная характеристика системы, отражающая степень ее соответствия потребностям и интересам ее заказчиков, пользователей, других заинтересованных лиц [6]. Специфика оценки эффективности информационных технологий состоит в отражении их результатов, которые не являются доходами от продаж продукции, товаров, работ и услуг.

Рассмотренный выше классический метод определения эффективности инвестиционных проектов применительно к информационным технологиям сопряжен с существенными сложностями по оценке результатов, эффектов от реализации, так как информационные технологии сами по себе не улучшают положение организации на рынке, не сокращают материалоемкость конечной продукции, а вооружают управленческий персонал новым оружием – технологиями [5].

Существующие методы оценки эффективности внедрения ИС автор статьи [9] подразделяет на следующие группы: традиционные финансовые методы, вероятностные и качественные. В рамках группы **фи-**

финансовых методов используют метод анализа затрат по видам деятельности (Activity Based Costing, ABC), метод расчета рентабельности инвестиций (Return on Investment, ROI) и метод расчета общей (совокупной) стоимости владения ИС (Total Cost of Ownership, TCO) наряду с общепринятыми финансовыми критериями (чистая дисконтированная стоимость, внутренняя норма прибыли).

В вероятностных методах используются статистические и математические модели, позволяющие оценить вероятность возникновения риска (например, повышение конкурентоспособности продукции, снижение рисков своевременного завершения проекта). Данные методы нужны для оценки будущего эффекта от применения информационной системы, но пока еще не так широко распространены в практике, как количественные и качественные. Полноценному использованию финансовых и вероятностных методов мешает невозможность в современных экономических условиях точно спрогнозировать изменение технико-экономических показателей работы предприятия (объем и продолжительность выпуска разрабатываемой продукции).

Качественные методы оценки, называемые также эвристическими, дополняют количественные расчеты, что может помочь оценить все явные и неявные факторы эффективности ИС и увязать их с общей стратегией компании.

Автором [2] предлагается выделить три типа эффектов, определяющих эффективность ИТ. Первый – эффект от учета дополнительной информации. Величина эффекта от автоматизации учета, как показывает практика, мала. Второй – эффект от нормирования и учета процессов более ощутим. Третий – эффект от планирования, оптимизации, управления процессами и ресурсами – самый значительный эффект, так как состоит в непосредственном сокращении ресурсов и ускорении процессов (при достижении тех же целей) за счет их «правильной организации».

К неявным факторам эффективности ИТ относятся: информированность руководства; снижение затрат труда на учет; уменьшение потерь от погрешностей учета; повышение точности и оперативности текущих управленческих процедур, предотвращение потерь; повышение управляемости, сохранение позиций на рынке, конкурентоспособность; снижение издержек от ошибок управления; улучшение взаимодействия с партнерами. Сложность состоит в экономической оценке этих выгод [3].

Анализ зарубежного опыта [8] свидетельствует о постоянном стремлении оценивать эффект от внедрения информационных систем по

различным показателям, основным среди которых является показатель ROI с точки зрения наглядности и простоты для руководителей компании и инвесторов. В российской практике заказчиков, как правило, интересует не количественная, а качественная оценка эффективности внедрения ИС. Работы по количественной оценке эффективности внедрения информационной системы – чрезвычайно трудоемкий и дорогой процесс.

В том случае, когда количественно проследить эффект от информационных технологий часто не представляется возможным, методом оценки эффективности информационных систем, выступает совокупная стоимость владения (Total Cost of Ownership, TCO). Эта модель предназначена для оценки сравнительной эффективности, так как основана на следующей предпосылке: две информационные системы характеризуются одинаковой эффективностью, и следует выбрать наименее затратную из них. Под совокупной стоимостью владения понимаются «полностью учтенные ежегодные расходы предприятия, связанные с приобретением и, что особенно важно, использованием информационных технологий в бизнесе».

По мере роста цивилизованности российских рыночных отношений, а также профессионализма российского менеджмента стали вырабатываться некоторые критерии оценки целесообразности затрат на внедрение ИТ. Наибольшее распространение получают показатель затрат как доля от оборота компании и показатель доли затрат на одного работающего. Показатель затрат как доли от оборота российских компаний составляет от 0,6% до 1,5% и существенно дифференцируются по отраслям.

В отсутствие единой, общепризнанной и универсальной методики процесс оценки эффективности ИС превращается в значительной степени в исследовательскую работу с необходимостью проектной привязки к конкретному объекту. В различных ситуациях должны применяться разные методы оценки эффективности. Оценка эффективности внедрения ИС должна носить комплексный характер, основанный как на количественных, так и на качественных оценках эффективности и учитывать все эффекты в каждом конкретном случае.

Литература

1. Бронникова Т. Оценка эффективности внедрения информационной системы управления предприятием. Измеримые цели и контроль их достижения / Т. Бронникова // Экономика и жизнь. – 2008. – № 47.
2. Гараева Ю. Найдены истоки эффективности ИТ [Электронный ресурс] / Ю. Гараева, Е. Фролов. – Режим доступа: <http://www.iteam.ru>
3. Гулиев Я.И. Оценка экономической эффективности использования информационных технологий в медицине [Электронный ресурс] / Я.И. Гулиев. – Режим доступа: <http://institutiones.com>

4. Зиндер Е. Что такое «эффективность ИТ»? [Электронный ресурс] / Е. Зиндер. – Режим доступа: www.fostas.ru
5. Оценить нельзя верить / А. Кадушин, Н. Михайлова // ИТ-форум. – 2003. – №5
6. Лопатников, Л.И. **Экономико-математический словарь**: словарь современной экономической науки. [Электронный ресурс] / Л. И. Лопатников – 5-е изд., перераб. и доп. – М.: Дело, 2003. – Режим доступа: <http://slovari.yandex.ru>
7. Провалов, В.С. Информационные технологии управления учеб. пособие – Москва: Флинта, 2008. – 376 с.
8. Романова, И.Ю. Современные подходы к оценке эффективности ИС управления [Электронный ресурс] / И.Ю. Романова, В.Л. Пархоменко. – Режим доступа: <http://economuch.com>
9. Ядыков, С. Эффективность информационных систем: докопаться до истины / С. Ядыков // Консультант. – 2010. – № 5.

УДК 004

МЕДИЦИНСКАЯ ИНТЕРАКТИВНАЯ ИНФОРМАЦИОННАЯ СИСТЕМА «MEDIACARE»: ОСНОВНЫЕ КОНЦЕПЦИИ, СИСТЕМНЫЕ И ТЕХНИЧЕСКИЕ РЕШЕНИЯ

Ковалевский Д.В., Широбокова С.Н.

Южно-Российский государственный технический университет (НПИ), г. Новочеркасск

Описана интерактивная информационная система «MediaCare», ее функциональная структура, системные решения, архитектурная система, основные функции и технические решения.

MediaCare [1] – интерактивная медицинская информационная система, обеспечивающая медицинским организациям основу для более высококачественного обслуживания пациентов благодаря применению новых наиболее эффективных способов предоставления услуг с использованием информационных технологий и процессов реинжиниринга.

MediaCare предоставляет возможность пациентам, проходящим стационарное лечение в палатах медицинской организации и находящимся там большую часть времени, взаимодействовать с персоналом медицинской организации на более высоком уровне, своевременно сообщая, посредством нажатия на соответствующую кнопку, о своих нуждах, либо, в критических ситуациях, о резком ухудшении состояния здоровья. Кроме того, MediaCare позволяет пациентам, в период стационарного лечения, не терять контакта с окружающим миром, открывая через терминалы, которыми оборудованы палаты, доступ к Internet, звонкам по телефонной линии, передаче сообщений (SMS) и Skype. Отдельное место в системе занимают модули, открывающие доступ к определенным видам контента, та-

ким как музыка, аудиокниги, радио, видео ролики, кинофильмы, видео игры. Пациент выбирает интересующий его контент либо из соответствующей категории, либо выбирает так называемый “пакет контента”, содержащий разнообразный, чаще всего тематический, контент. Остановившись в своем выборе на платном контенте, пациент может оплатить его как с помощью кредитной карты, так и через систему PayPal или путем предоплаты. Кроме всего прочего, предусмотрены клиентские приложения администраторов и медсестер, через которые можно получить доступ к функциям администрирования. Схема функциональной структуры MediaCare представлена на рис. 1.

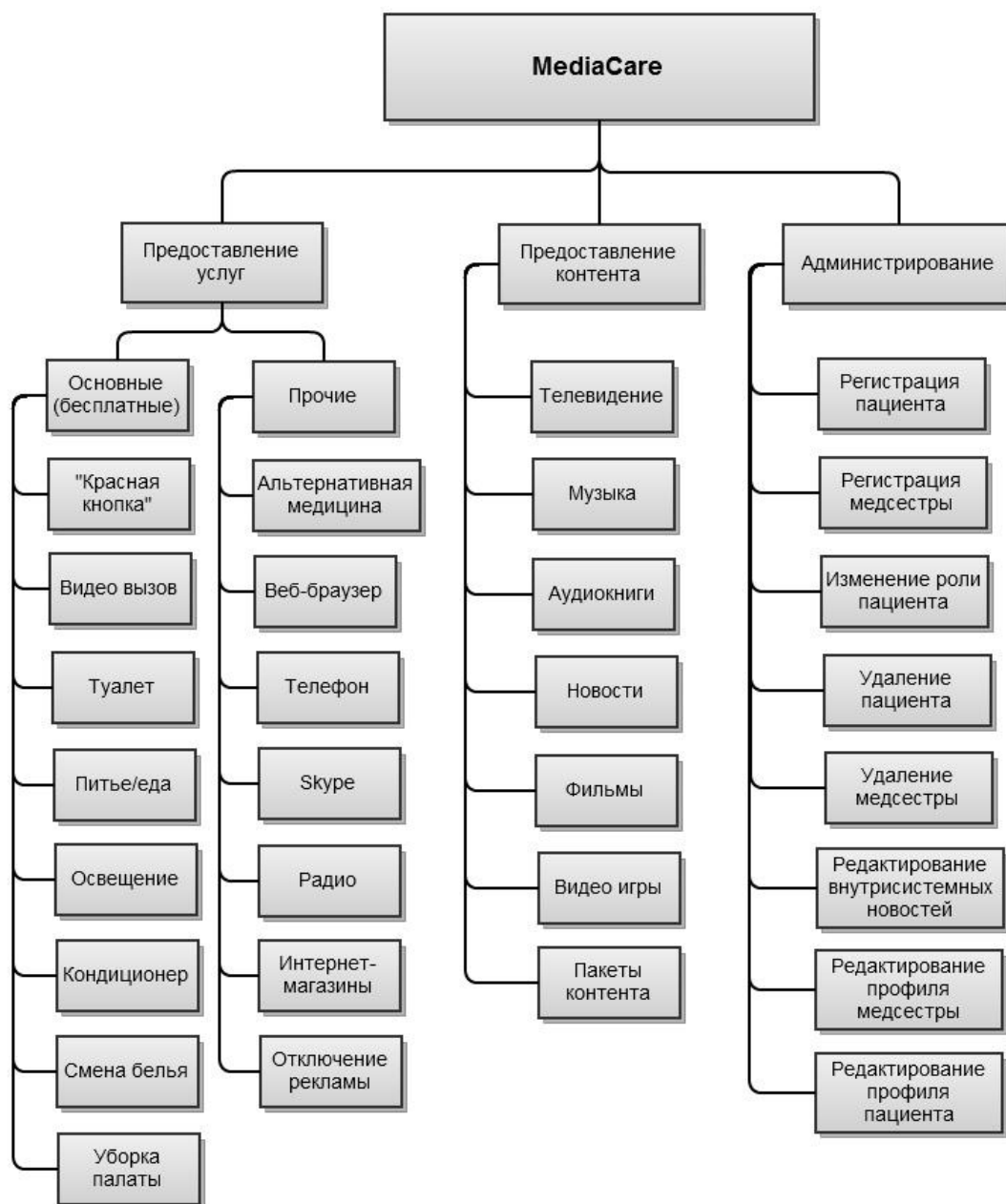
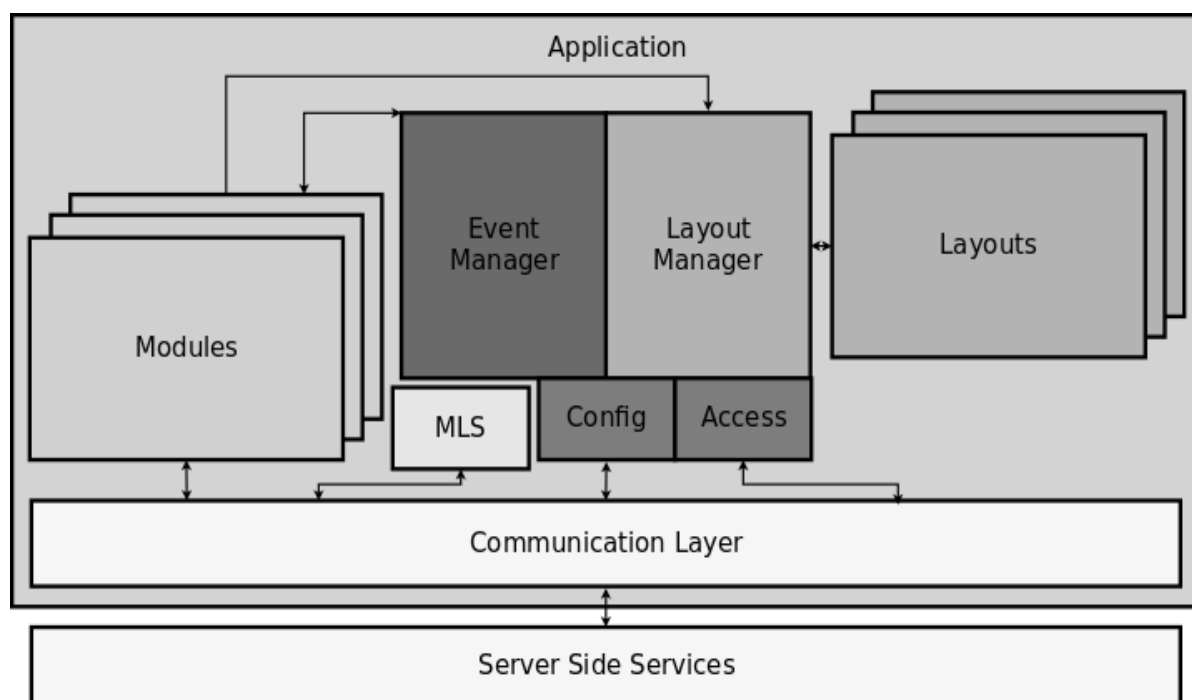


Рис. 1. Схема функциональной структуры MediaCare

Информационная система MediaCare разработана в клиент-серверной архитектуре, где клиентская часть представляет собой веб-приложение, перенесенное на специализированный сенсорный ПК (терминал), размещенный в палате пациента, в то время как серверная – установлена на ПК, выполняющем функции сервера.

Серверная часть системы MediaCare состоит из сервера приложения, на котором располагается база данных системы, обеспечивающего работу с множеством клиентов (как пациентов, так и медсестер и администраторов), и видео-сервера, необходимого для обеспечения потокового видео вещания VLC (одна из услуг, доступная пациенту). В качестве аппаратной базы для обоих серверов используются персональные компьютеры.

Системную архитектуру всего приложения можно представить схематически в виде, представленном на рис. 2 [1].



H-Portal Client and Nurse application architecture

Рис. 2. Архитектура MediaCare

Modules – модули MediaCare, выполняющие каждый свою определенную функцию, связанную с предоставлением услуг или контента пациентам.

Event Manager – обработчик событий, отвечающий за точное соблюдение программной логики и правильной последовательности выполнения операций.

Layouts – макеты страниц графического интерфейса MediaCare с указанием мест расположения блоков, в которых отображается информация различного характера.

Layout Manager – менеджер макетов, отвечающий главным образом за их отрисовку (рендеринг) и скрытие.

MLS – подсистема управления языковыми средствами системы.

Access – подсистема управления доступом к услугам и контенту.

Config – подсистема конфигурирования системы в зависимости от типа и категории пользователя.

Communication Layer – слой взаимодействия клиентского и серверного приложений.

Server Side Services – серверная часть MediaCare.

На рис. 3 [1] представлена диаграмма потоков данных в клиентском приложении пациента MediaCare, достаточно подробно отражающая принципы работы медицинской интерактивной информационной системы.

Ввиду того, что информационная система предоставляет как платные, так и бесплатные услуги и контент, встает вопрос о разграничении доступа.

Для выполнения функции разграничения доступа выполняются следующие условия:

- Каждый пациент должен принадлежать к определенной личной категории (Personal Category). Это категория должна создавать при регистрации пациента и не должна изменяться или удаляться.

- Личная категория ассоциируется с доступными данному пациенту услугами и контентом. Данные взаимосвязи используются для того, чтобы разрешать или запрещать доступ к ним.

- Личные категории имеют высший приоритет. Если один контент недоступен какой-либо пользовательской категории (User Category), но доступен личной, то доступ к нему должен быть разрешен.

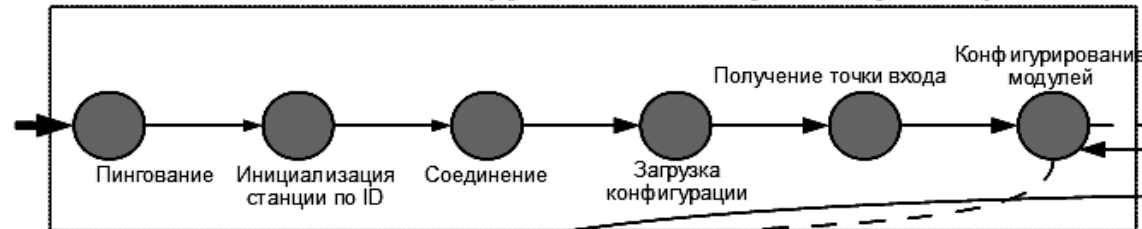
- Весь платный и не оплаченный контент имеет указывающее на это свойство “Payment” (оплата) со значением “Sale” (платный). Для того чтобы получить доступ к нему необходимо совершить процесс оплаты.

- Весь оплаченный контент имеет свойство “Payment” (оплата) со значением “Paid” (оплачено). Весь бесплатный контент имеет свойство “Payment” (оплата) со значением “Free” (бесплатный).

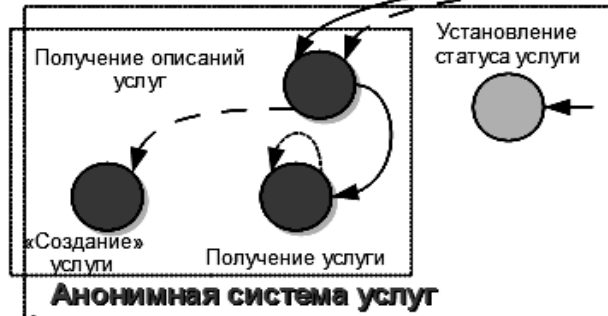
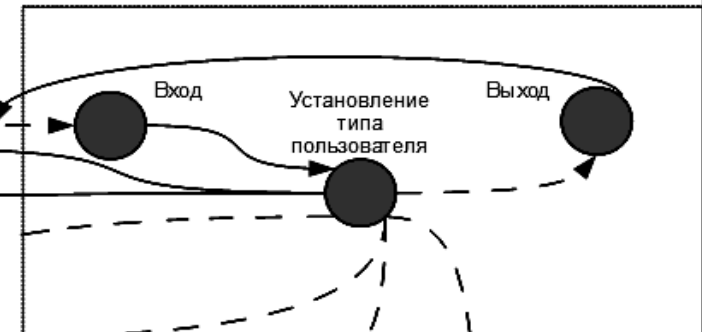
Диаграмма потоков данных в клиентском приложении пациента системы MediaCare

- > Функция обратного вызова
- -> Функция становится доступной для вызова

Инициализация станции и загрузка для последующей аутентификации

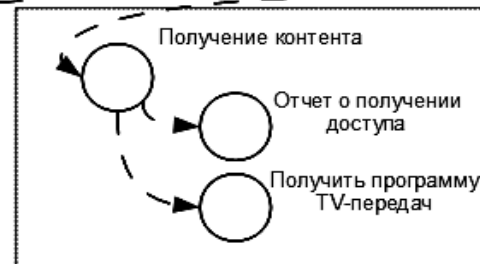
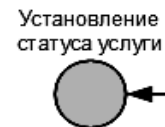


Вход и выход

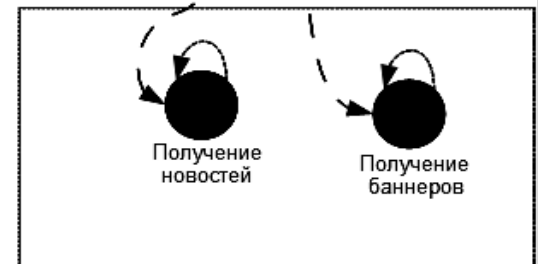


Анонимная система услуг

Система услуг пациента



Работа с контентом



Основной контент

Анонимная система услуг получает услуги, созданные на сервере, а система услуг пациента получает услуги «созданные» пациентом на своей станции. Это разные услуги!

Рис. 3. Диаграмма потоков данных в клиентском приложении пациента системы MediaCare

Процесс оплаты должен происходить следующим образом. Если пациент решает оплатить услугу, либо контент, то, при нажатии на соответствующую кнопку, отображается диалоговое окно оплаты услуги (контента) предлагающее выбрать тип оплаты, внешнюю платежную систему и ввести все необходимые данные для совершения платежа. После ввода данных и подтверждения операции совершения платежа, запрос, содержащий платежные реквизиты пациента, а также данные об оплачиваемой услуге, отправляется на сервер внешней платежной системы. В случае успешной отправки запроса и подтверждения факта платежеспособности, происходит списание средств с кредитной карты пациента, изменение роли пациента в системе, в результате чего оплаченная услуга (контент) становится доступна ему. Отчет о проделанной операции выводится на экран, а данные о совершенном платеже фиксируются в базе данных. Важным замечанием является то, что платежные реквизиты пациента не остаются в системе в каком бы то ни было виде, они лишь передаются внешней платежной системе, которая в свою очередь обеспечивает необходимый уровень безопасности в ходе процесса.

Для проектирования клиентских приложений медицинской системы использована платформо-независимая среда Adobe Air [2]. Доступ к объектам приложения осуществляется, используя сценарии, написанные на объектно-ориентированном скриптовом языке программирования JavaScript с использованием распространенной библиотеки для данного языка JQuery, значительно упрощающей процесс программной реализации за счёт возможности получения доступа к любому элементу интерфейса, его атрибутам и содержимому.

Серверная часть системы реализована на объектно-ориентированном языке программирования C# с использованием ECO (Enterprise Core Objects) – программной платформы предназначенной для доменно-ориентированной разработки (парадигма объектно-ориентированного программирования, нацеленная на ускорение разработки программных проектов, связанных со сложными доменами). ECO позволяет повысить производительность за счет использования средств, таких как ORM (англ. Object-relational mapping, рус. Объектно-реляционное отображение), UML-модели классов предметной области. В качестве системы управления базами данных была выбрана MySQL – свободно-распространяющаяся СУБД от компании Oracle. Доступ к базам данных MySQL осуществляется через функционал ECO.

Все функциональные подсистемы медицинской интерактивной информационной системы MediaCare следуют архитектурной схеме «Модель-представление-контроллер» (MVC).

Взаимодействуя с приложением, реализованным на JavaScript [3], через форму, являющуюся в данном контексте *представлением*, браузер отправляет *запрос* «обработчику событий» (Event Manager), одному из функциональных модулей системы, который по своей сути является *контроллером* и отвечает за определение того, какое дальнейшее действие в системе повлечет за собой то или иное «событие», произошедшее в ответ на действие пользователя. В некоторых случаях контроллер сразу визуализирует *представление*, обрабатывая входные данные, конвертируя результат в HTML и отсылая его в систему, однако в большинстве случаев *контроллер* взаимодействует с *моделью*, которая представлена объектом ЕСО (Enterprise Core Objects) [4], отвечающим за коммуникацию с базой данных. После вызова и изменения *модели*, *контроллер* также визуализирует *представление*, содержащее результат изменения *модели*, и возвращает готовую веб-страницу в систему как HTML.

За этапами функционирования архитектурной схемы MVC [5], внедренной в платежную подсистему можно проследить на рис. 4.

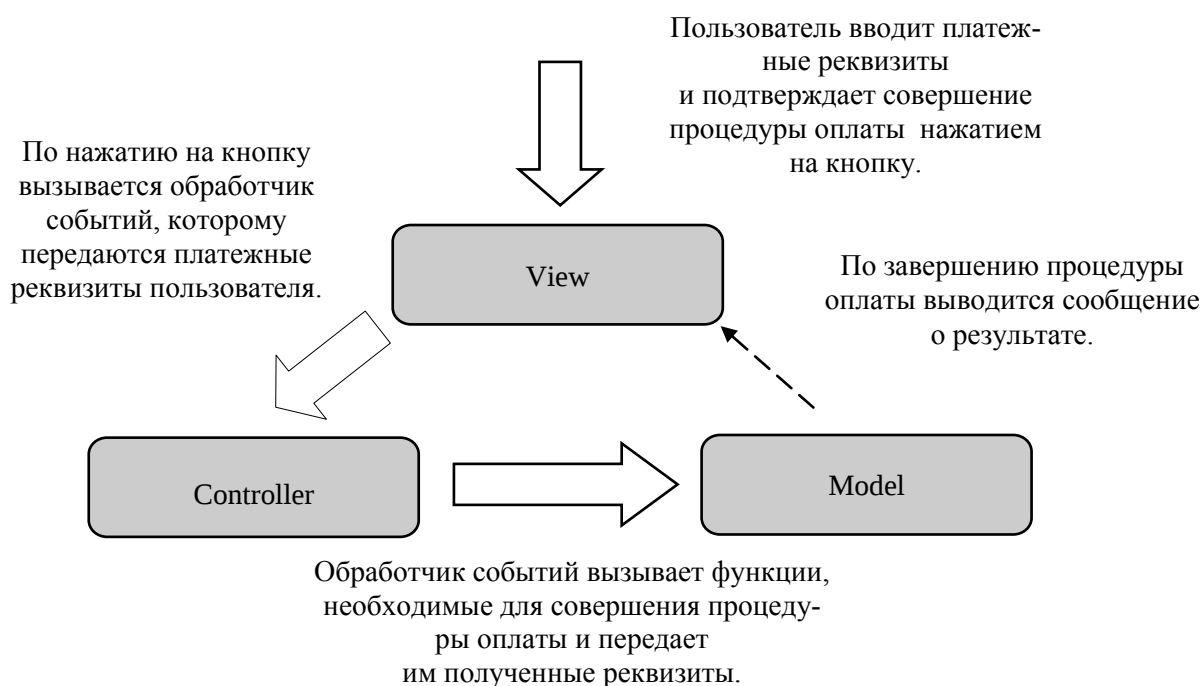


Рис. 4. Архитектурная схема MVC в платежной подсистеме медицинской интерактивной информационной системы MediaCare

В нашем примере *представлением* являются диалоговое окно оплаты и все связанные с ним окна платежной подсистемы, роль *контроллера* играет вышеупомянутый «обработчик событий», а роль *модели* принадлежит сразу нескольким функциональным модулям, содержащим методы работы с данными платежной подсистемы.

Таким образом, применение архитектурной схемы MVC в проектировании информационной системы позволяет максимально разделить бизнес-логику от ее визуализации, тем самым способствуя более эффективной организации доступа к каждому модулю MediaCare из любого другого.

Система управления версиями (от англ. Version Control System, VCS или Revision Control System) – программное обеспечение для облегчения работы с изменяющейся информацией. Система управления версиями позволяет хранить несколько версий одного и того же документа, при необходимости возвращаться к более ранним версиям, определять, кто и когда сделал то или иное изменение, и многое другое.

С целью обеспечения совместной разработки, а так же для контроля версий проекта с возможностью многократных коммитов в ответвленную часть от основной ветки проекта, используется систему управления версиями GitHub. GitHub – самый большой веб-сервис для хостинга проектов и их совместной разработки. Основан на системе контроля версий Git (распределённая система управления версиями файлов) и разработан на Ruby on Rails компанией Logical Awesome.

Сервис абсолютно бесплатен и предоставляет все возможности (включая SSL) для проектов с открытым исходным кодом, а для частных проектов предлагаются различные платные тарифные планы.

Кроме размещения кода, участники могут общаться, комментировать правки друг друга, следить за новостями знакомых. С помощью широких возможностей Git программисты могут объединять свои репозитории (директории, содержащие исходный код на клиентском компьютере). GitHub предлагает удобный интерфейс для этого и может отобразить вклад каждого участника в виде дерева.

Часто бывает, что над одним проектом одновременно работают несколько человек. Если два человека изменяют один и тот же файл, то один из них может случайно отменить изменения, сделанные другим. Системы управления версиями отслеживают такие конфликты и предлагают средства их решения. Большинство систем может автоматически объединить (слить) изменения, сделанные разными разработчиками. Однако такое автоматическое объединение изменений, обычно, возможно только для тек-

стовых файлов и при условии, что изменялись разные (непересекающиеся) части этого файла. Такое ограничение связано с тем, что большинство систем управления версиями ориентированы на поддержку процесса разработки программного обеспечения, а исходные коды программ хранятся в текстовых файлах. Если автоматическое объединение выполнить не удалось, система может предложить решить проблему вручную.

В случае если работа сервера прекращена, через который шла работа, любой клиентский репозиторий может быть скопирован обратно на сервер с целью восстановления кода. Каждый раз, когда клиент “забирает” свежую версию файлов, создаётся полная копия всех данных (рис. 5).

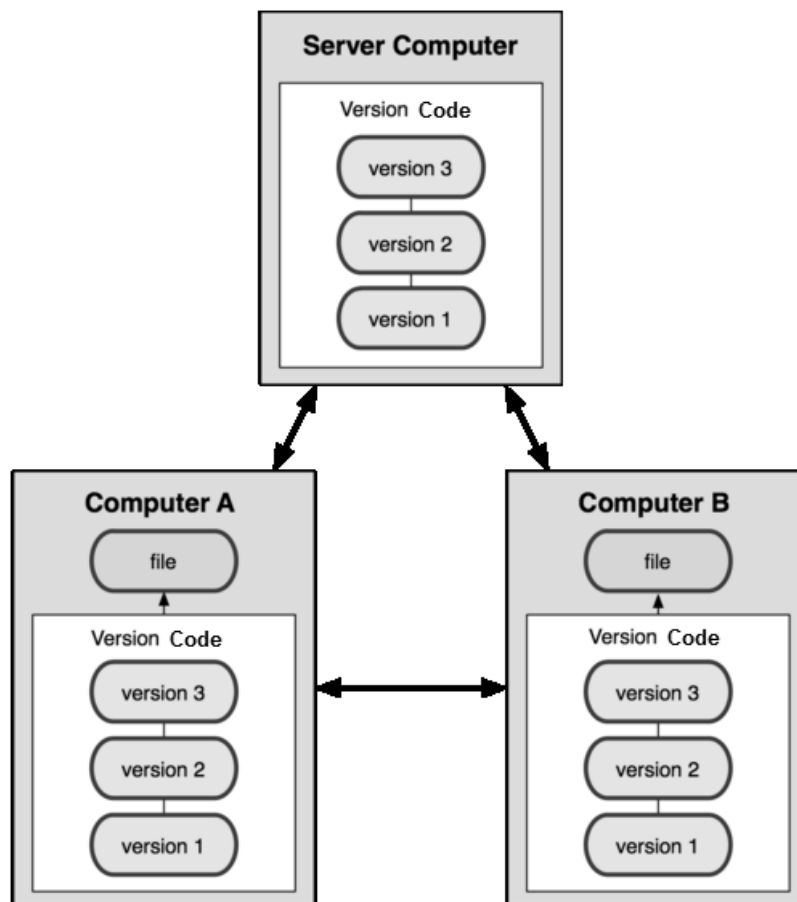


Рис. 5. Схема распределенной системы управления версиями

Данное сочетание программных средств разработки позволило в полной мере реализовать функционал медицинской информационной системы, обеспечить создание интуитивно-понятного интерфейса для любой возрастной группы пациентов и обеспечить необходимый уровень безопасности при выполнении процесса совершения платежа на всех его этапах.

Тестирование является неотъемлемой частью разработки программного обеспечения.

Автоматизированное тестирование программного обеспечения – часть процесса тестирования на этапе контроля качества в процессе разработки программного обеспечения. Оно использует программные средства для выполнения тестов и проверки результатов выполнения, что помогает сократить время тестирования и упростить его процесс. То есть на основе тестовой документации (тестовые сценарии, тестовые случаи, тестовые модели) создается решение по автоматизации, которое должно обеспечивать максимальное покрытие тестами функционала разрабатываемого продукта.

Пример реализации тестового сценария:

Фрагменты документа Test-plan представлены в табл. 1, 2 и 3.

Таблица 1

Открытие телевизионного модуля

ID	MCS100.3.1-1. Open TV screen.
Actions	1. Log in a system. 2. Click 'Activities' button.
Expected Result	'TV' screen with saving previous position is opened.
Actual Result	1. 'TV' screen with saving previous position is opened. 2. 'TV' screen isn't opened. 3. 'TV' screen without saving previous position is opened.

Таблица 2

Открытие окна выбора услуг

ID	MCS100.3.2-1. Show Activities window.
Actions	1. Log in a system. 2. Click 'Activities' button.
Expected Result	Activities window is opened.
Actual Result	1. Activities window is opened. 2. Activities window isn't opened.

Вызов медсестры

ID	MCS100.3.3-1. Call a nurse.
Actions	1. Log in a system. 2. Click Red Button.
Expected Result	Nurse is called.
Actual Result	1. Nurse is called. 2. Nurse isn't called.

ID – Идентификационный номер и описание тестового случая.

Actions – последовательность действий инженера выполняющего тест-план.

Expected Result – ожидаемый результат выполнения действий.

Actual Result – возможный результат выполнения действий.

Технически MediaCare строится на базе локальной вычислительной сети, где роль сервера и рабочих станций медсестер и администраторов системы играет персональный компьютер, а рабочих станций пациентов – специализированный больничный терминал Advantech PIT-1502 (рис. 6.). Технические характеристики сервера и терминалов указаны в табл. 3 и 4.



Рис. 6. Больничный терминал Advantech PIT-1502

Таблица 4

Технические характеристики типового сервера MediaCare

Процессор	Intel Xeon E5630 2.53/5.86GTsec/12M LGA1366 AT80614005463AA (Gulftown)
Материнская плата	INTEL S5520HC (Hanlan Creek) Boxed Intel® Server Board S5520HC (12 DIMMs)
Оперативная память	Kingston KVR1333D3S4R9S/2G ECC REG DDR-III – 3 шт.
Сетевая карта	Intel E1G42ET
HDD	HDD Server SEAGATE Barracuda ES 500Gb 7200.2 ST3500320NS SATA 32 mb
DVD±RW-привод	Optiarc AD-5200S-OB
Программное обеспечение	Linux Ubuntu

Таблица 5

Технические характеристики Advantech PIT-1502

Процессор	CPU Intel PineView (N450/D510) 1.6 GHz
Материнская плата	System Chipset Intel N450/D510 + ICH8M
Оперативная память	RAM 2GB
Сетевая карта	LAN 10/100/1000 RJ45 x 1, WLAN 802.11 b/g/n
HDD	Storage CF or SATA HDD interface
Дисплей	15.6" W TFT
Программное обеспечение	Linux Ubuntu

Как сервер, так и все клиентские станции работают под управлением операционной системы Linux Ubuntu. Данный выбор был продиктован повышенными требованиями к надежности и безопасности функционирования системы, а так же возможностью более гибкого конфигурирования терминалов пациентов ввиду особенностей системной архитектуры.

Схему структуры типовой локальной сети для обеспечения полноценной работы всех групп пользователей MediaCare можно изобразить следующим образом (рис. 7.)

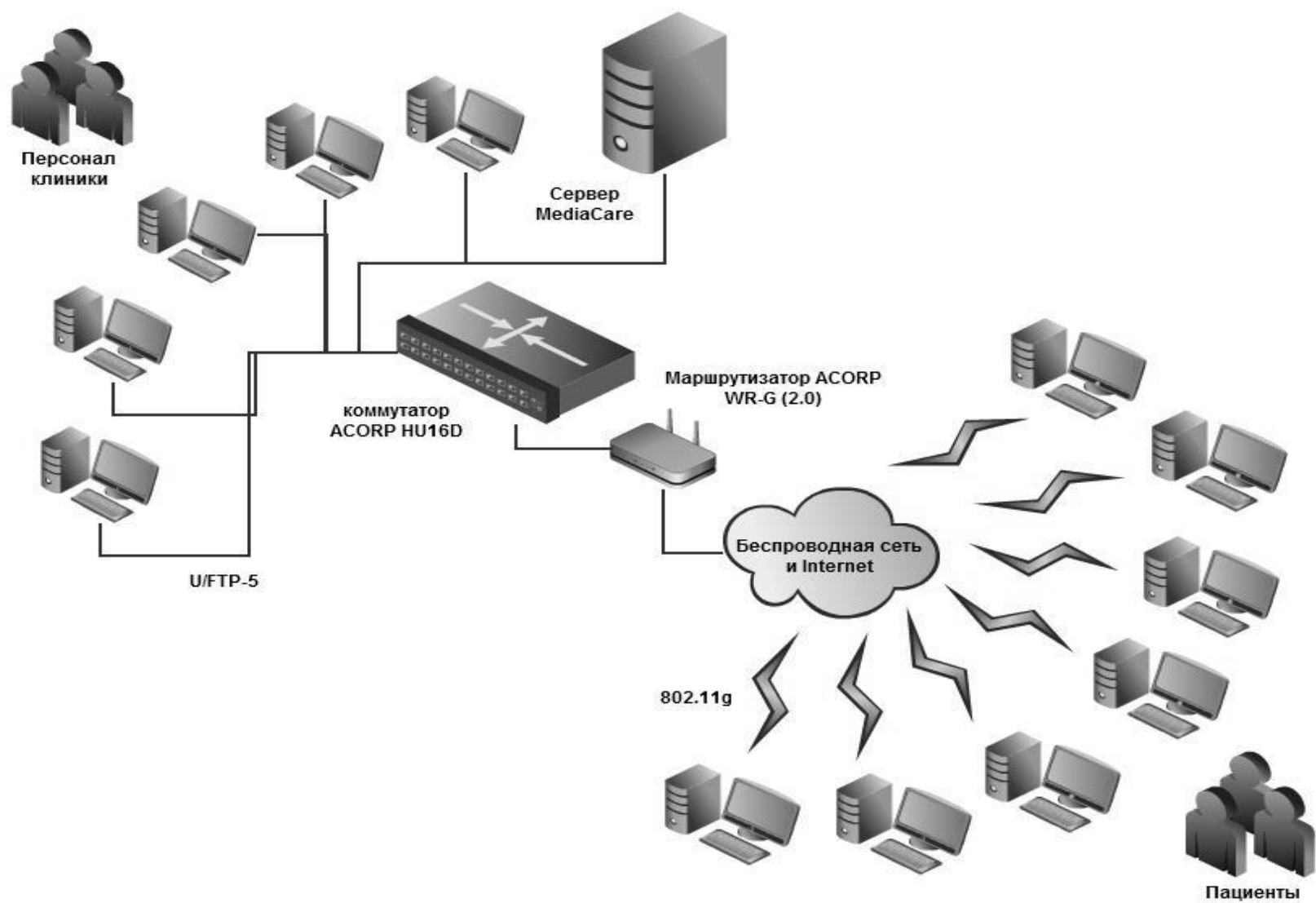


Рис. 7. Схема структуры типовой локальной сети MediaCare

Взаимодействие клиентского приложения с серверной частью исключительно по беспроводной сети Wi-Fi стандарта 802.11g обусловлено требованиями к мобильности, во многом по той причине, что ведется разработка версии MediaCare для мобильных платформ, снижением издержек на прокладку кабеля в каждую палату и к каждому терминалу, гибкостью в построении и гарантом легкости и экономичности расширения сети.

Связь между рабочими станциями персонала клиники с сервером организуется традиционно с использованием кабеля U/FTP-5 и коммутатора (в данном примере – ACORP HU16D).

Как отмечалось выше, внедрение медицинской интерактивной информационной системы MediaCare в клиниках закладывает основу для более высококачественного обслуживания пациентов благодаря применению новых наиболее эффективных способов предоставления услуг с использованием информационных технологий и процессов реинжиниринга. Использование в совокупности всех вышеизложенных концепций, а также системных и технических решений при проектировании информационной системы дало веское основание для данного утверждения. Медицинское учреждение с MediaCare, внедренной в инфраструктуру, имеет все шансы как минимум укрепить свое положение на рынке медицинских услуг, предложив при этом своим пациентам максимально комфортные условия нахождения в палатах при стационарном лечении.

Литература

1. ООО “Прог-Форс”, MediaCare Application Documentation, 2011.
2. Джои Лотт, Кэтрин Ротондо, Сэмюел Ан, Эшли Аткинс. Adobe AIR. Практическое руководство по среде для настольных приложений Flash и Flex. – Символ-Плюс, 2009.
3. Дэвид Флэнаган. JavaScript. Подробное руководство. – Символ-Плюс, 2008.
4. Delphi Wikia: Enterprise Core Objects
http://delphi.wikia.com/wiki/Enterprise_Core_Objects.
5. Стивен Сандерсон. ASP.NET MVC Framework с примерами на C# для профессионалов. – М.: Вильямс, 2009. – ISBN 978-5-8459-1609-9.

УДК 004

КОНЦЕПЦИИ И ВОПРОСЫ ПРОЕКТИРОВАНИЯ ИНФОРМАЦИОННЫХ СИСТЕМ АБОНЕНТСКОГО ОБСЛУЖИВАНИЯ НАСЕЛЕНИЯ ДЛЯ ПРЕДПРИЯТИЙ ВОДОСНАБЖЕНИЯ И ВОДООТВЕДЕНИЯ

Ковалевский В.Н.

Южно-Российский государственный технический университет (НПИ), г. Новочеркасск

Рассмотрены вопросы концептуального проектирования биллинговой информационной системы абонентского обслуживания населения для водоснабжающих

предприятий. Приведены основные алгоритмы расчетов за услуги и функциональные модули их реализации. Отмечено практическое применение проектных решений.

Расчеты с абонентами за услуги водопотребления и водоотведения на предприятиях водо-канализационного хозяйства являются весьма непростой задачей. Особенно это касается обслуживания физических лиц (населения). Сложность расчетов обусловлена несколькими факторами. Один из них – отсутствие приборов учета в большом секторе проживания абонентов. Как следствие, для определения объема оказываемых услуг приходится использовать достаточно сложные алгоритмы, опирающиеся на потребительские свойства жилья (комфортность), количество проживающих, количество единиц дополнительного потребления (полив, автотехника, домашние животные). При наличии у абонентов приборов учета возникают другие сложности в алгоритмах начисления объемов потребленных ресурсов, а именно, появление различных вариантов конфигурации установки водосчетчиков в доме. Особенно это касается многоэтажных домов, коммунального жилья и домов с наличием (как правило, на первом этаже) юридических организаций. На таких объектах необходимо устанавливать общий на вводе дома водосчетчик и разрабатывать различные схемы подключения абонентов и субабонентов. Другим фактором сложности расчетов с населением является большое количество абонентов (тысячи или десятки тысяч) у одного водоснабжающего предприятия. Поэтому в настоящее время абонентское обслуживание физических лиц вручную или в полуавтоматическом режиме (например, использование файлов MS Excel, таблиц MS Access) трудоемко и практически невозможно. Для обеспечения более высококачественного обслуживания абонентов необходимо применение новых технологий на базе биллинговых информационных систем [1]. Вопросам концептуального проектирования таких автоматизированных систем посвящена данная работа.

При анализе предметной области можно выделить следующие функции абонентского обслуживания физических лиц (рис. 1).

Эти функции, как видно из рисунка, разбиваются на три основные группы. Первая группа “Работа с картотекой абонентов” используется во время первоначальной постановки абонентского учета на ресурсоснабжающем предприятии, а также для существенной корректировки данных (передача предприятию новых участков, улиц, домов и соответственно их удаление из обслуживания).

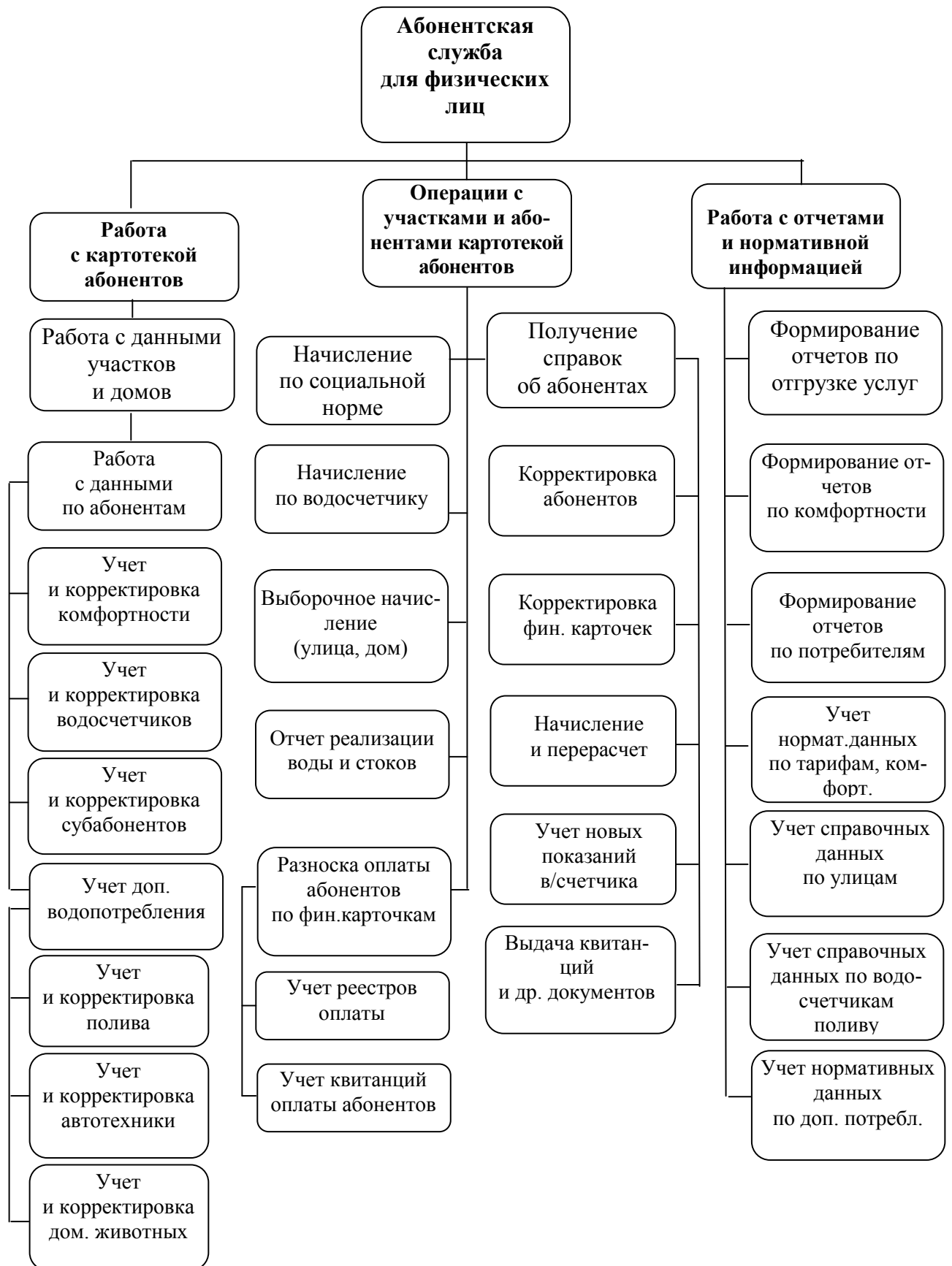


Рис. 1. Функции абонентской службы

Группа функций “Операции с участками и абонентами” охватывает текущую работу персонала абонентского отдела:

- прием посетителей по различным вопросам, включающим выдачу сведений о задолженности (переплате) абонентов, корректировку данных по абоненту (количество проживающих, изменение комфортности (удобств), замена водосчетчика) и соответственно перерасчет объемов и стоимости услуг за различный период, а также выдача квитанций, актов и других документов;

- учет новых показаний по квитанциям, полученным от абонентов;

- корректировка финансовых карточек абонентов по их заявлениям;

- ежемесячное начисление абонентам объемов оказанных услуг по воде, стокам и их стоимости в разрезе участков, занесение этих данных в финансовые карточки;

- выборочное начисление абонентам по заданным улицам, домам;

- ежемесячное формирование отчета по реализации услуг предприятием в разрезе участков (города), групп потребителей;

- ежедневная разноска оплаты по квитанциям, поступающим от абонентов через банковские учреждения, почту России и кассу предприятия, и занесение этих данных в финансовые карточки;

- сверка по реестрам оплат полученных денег с финансовыми средствами, поступившими на расчетный счет предприятия;

- ведение финансовых карточек абонентов и расчет сальдо на начало следующего месяца.

Последняя группа “Работа с отчетами и нормативной информацией” относится сервисным функциям. Формирование отчетов и различных справок по деятельности абонентской службы выполняются персоналом по мере необходимости, согласно нормативно-законодательной документации, а также по запросу руководства предприятия. Нормативно-справочная информация должна постоянно находиться в актуальном состоянии и своевременно корректироваться руководством абонентской службы при выходе в печать соответствующих постановлений и указов.

Детальный анализ предметной области, позволивший получить перечень выше описанных функций абонентской службы, дает также возможность построить концептуальную модель функционирования рассматриваемой системы. Наиболее распространенным средством ее реализации являются инфологические диаграммы “сущность-связь”[2]. С их помощью определяются важные для предметной области объекты (сущности), их свойства (атрибуты) и отношения друг с другом (связи). Основная (первая) диаграмма абонентской службы для физических лиц представлена на рис. 2.

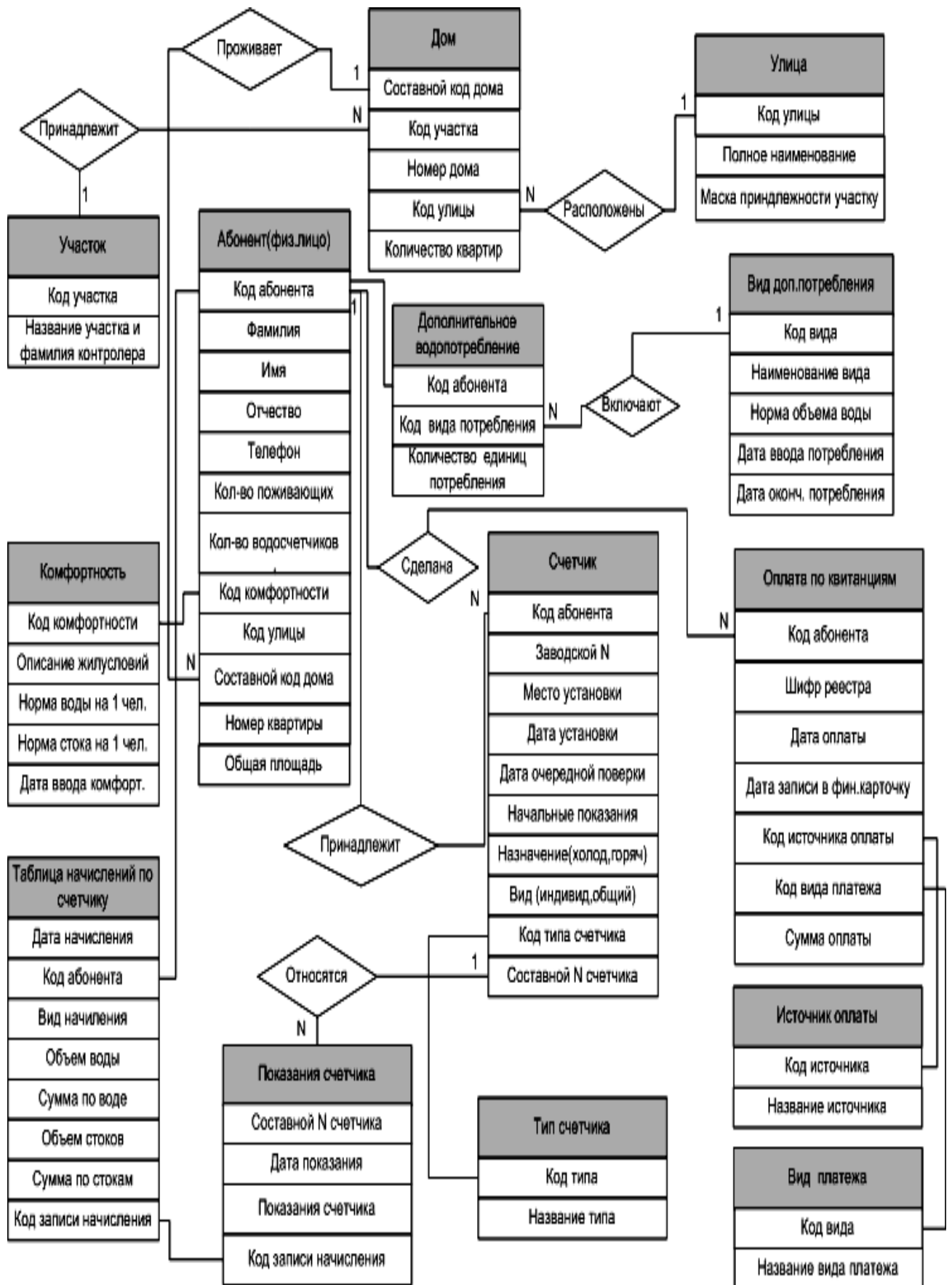


Рис. 2. Концептуальная модель абонентской службы (первая диаграмма)

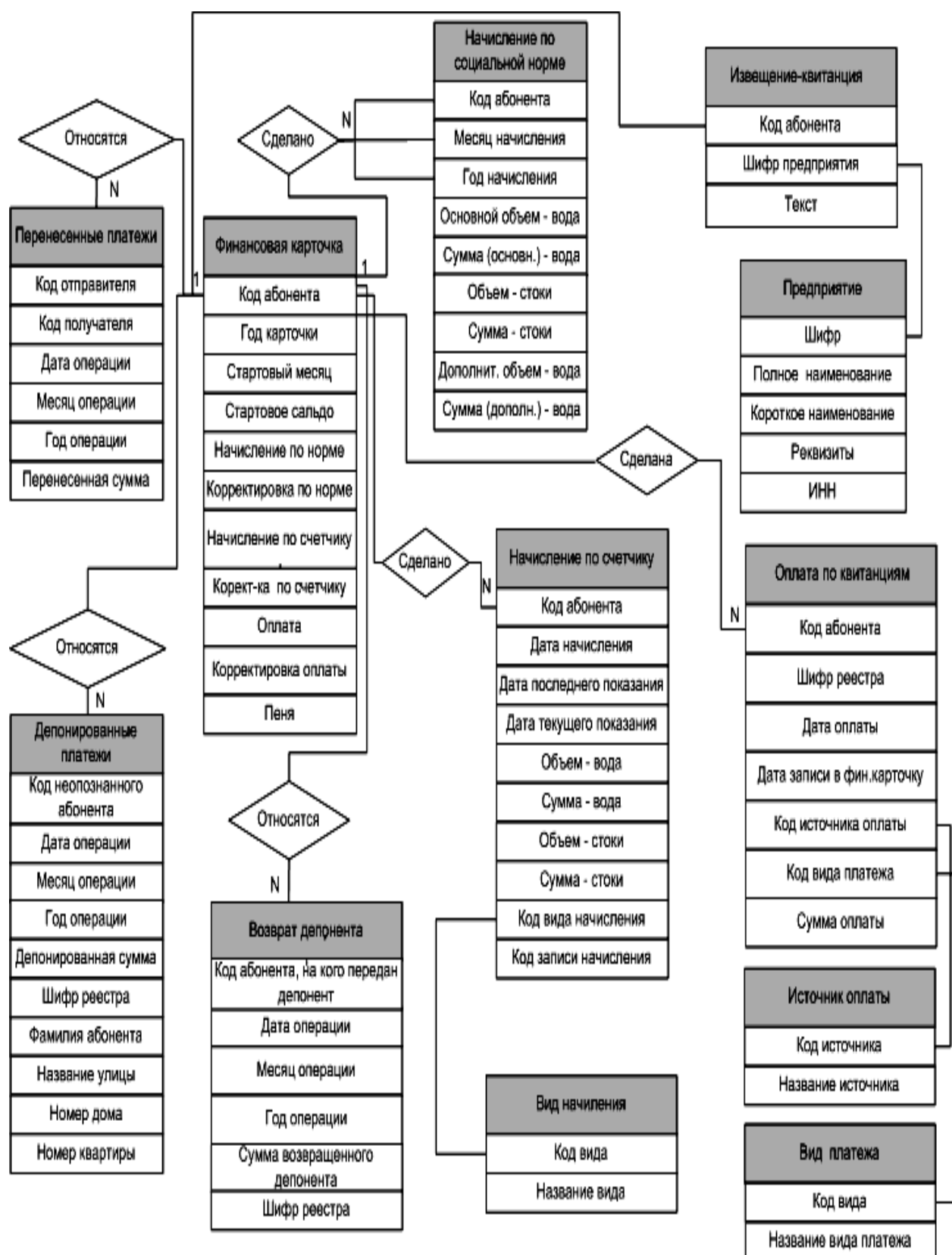


Рис. 3. Концептуальная модель абонентской службы (вторая диаграмма)

Основным объектом, с которым связано большинство других сущностей модели, является “Абонент (физическое лицо)”. Ключевой атрибут,

по которому осуществляется связь, называется “Код абонента”. Поэтому он должен быть уникальным.

Вторая диаграмма абонентской службы для физических лиц включает следующие объекты: «Финансовая карточка абонента», «Перенесенные платежи», «Депонированные платежи», «Возврат депонента», «Оплата по квитанциям», «Начисление по счетчику», «Начисление по социальной норме», «Извещение – квитанция» (рис. 3). Отношение сущности “Финансовая карточка” к объекту “Абонент (физическое лицо)” выражается как $N : 1$, где N – количество финансовых карточек за каждый год функционирования абонентской службы. Ключевым атрибутом связи всех выше перечисленных объектов также является уникальное свойство “Код абонента”.

Для полного формирования концептуальной модели абонентской службы потребовалось около **60** объектов. Все непоказанные на рисунках сущности предназначены для поддержки функций, связанных уже с непосредственной работой проектируемой информационной системы абонентской службы: внутреннее хранение отчетов и различных массивов данных, администрирование и защита системы, открытие и закрытие операционных периодов, взаимодействие с внешними информационными системами.

После выполнения построения концептуальной модели важными вопросами проектирования является определение алгоритмического состава системы, автономных и общих функциональных модулей. Анализ функций абонентской службы позволил выделить следующие компоненты:

- 1) модуль начисления услуг по социальной норме в разрезе абонента, дома, группы домов, улицы, участка;
- 2) модуль начисления услуг по водосчетчикам в разрезе абонента, дома, группы домов, улицы, участка;
- 3) модуль начисления услуг по общему водосчетчику для многоквартирных домов;
- 4) модуль автоматической обработки платежей абонентов по реестрам и их разности в финансовые карточки;
- 5) модуль потоковой печати извещений-квитанций абонентам в разрезе дома, группы домов, улицы, участка;
- 6) модуль формирования данных по реализации оказанных услуг в разрезе абонента, дома, группы домов, улицы, участка за различный период;
- 7) модули формирования данных по комфортности, группам потребителей в разрезе участка, города за различный период;
- 8) модули формирования файлов по потребленным и оплаченным услугам абонентов для различных социальных учреждений (субсидии, материальная помощь);

9) функциональные модули, встроенные в различные отчеты, связанные с текущей деятельностью абонентской службы;

10) модули, обеспечивающие открытие, сопровождение и закрытие операционных месяца, года.

Первый представленный модуль реализует алгоритм начисления объемов оказанных услуг по воде и стокам для абонентов, не имеющих счетчики:

$$V_{\text{норм_вод}} = N_{\text{ж_абон}} \times S_{\text{вод}}, \quad V_{\text{норм_сток}} = N_{\text{ж_абон}} \times S_{\text{сток}},$$

где $V_{\text{норм_вод}}$, $V_{\text{норм_сток}}$ – объем воды (стоков) по социальной норме за месяц; $S_{\text{вод}}$, $S_{\text{сток}}$ – объем социальной нормы воды (стоков) на человека за месяц; $N_{\text{ж_абон}}$ – количество человек, проживающих в квартире (одноквартирном доме).

Если у абонента имеется дополнительное потребление: полив, мойка автотехники, расход воды на домашних животных (корова, свинья, козы и т.д.), то объем воды по норме определяются:

$$V_{\text{норм_вод}} = N_{\text{ж_абон}} \times S_{\text{вод}} + N_{\text{пол}} \times S_{\text{пол}} + N_{\text{авт}} \times S_{\text{авт}} + N_{\text{жив}} \times S_{\text{жив}},$$

где $N_{\text{пол}}$ – площадь полива, $N_{\text{авт}}$ – количество автотехники, $N_{\text{жив}}$ – количество животных; $S_{\text{пол}}$ – норма воды на единицу площади полива, $S_{\text{авт}}$ – норма воды на единицу автотехники, $S_{\text{жив}}$ – норма воды на одного животного.

Объем стоков по норме тот же: $V_{\text{норм_сток}} = N_{\text{ж_абон}} \times S_{\text{сток}}$.

Второй модуль реализует алгоритм начисления объемов оказанных услуг по воде и стокам для абонентов, имеющих счетчики. Если у абонента отсутствует дополнительное потребление, то объемы по воде и стокам определяется:

$$V_{\text{сч_вод}} = V_{\text{сч}} \times (S_{\text{вод}} \div S_{\text{сток}}), \quad V_{\text{сч_сток}} = V_{\text{сч}} \times (S_{\text{сток}} \div S_{\text{вод}}),$$

где $V_{\text{сч_вод}}$, $V_{\text{сч_сток}}$ – объем воды (стоков) по счетчику за месяц; $V_{\text{сч}}$ – расход по показанию счетчика за месяц.

Если у абонента имеется дополнительное потребление, то алгоритм расчета выглядит следующим образом. Подсчитывается весь объем услуг по норме $S_{\text{общ}}$, включая дополнительное водопотребление, но без объемов по стокам:

$$S_{\text{общ}} = N_{\text{ж_абон}} \times S_{\text{вод}} + N_{\text{пол}} \times S_{\text{пол}} + N_{\text{авт}} \times S_{\text{авт}} + N_{\text{пол}} \times S_{\text{пол}}$$

Объем оказанных услуг по воде будет равен объему, показанному по счетчику: $V_{\text{сч_вод}} = V_{\text{сч}}$. В этот же объем будут входить объемы воды по дополнительному водопотреблению:

$$V_{сч_пол} = V_{сч} \times ((N_{пол} \times S_{пол}) \div S_{общ});$$

$$V_{сч_авт} = V_{сч} \times ((N_{авт} \times S_{авт}) \div S_{общ});$$

$$V_{сч_жив} = V_{сч} \times ((N_{жив} \times S_{жив}) \div S_{общ}).$$

Тогда объем оказанных услуг по стокам будет равен:

$$V_{сч_сток} = (V_{сч} - V_{сч_пол} - V_{сч_авт} - V_{сч_жив}) \div (S_{сток} \div S_{вод}).$$

Последняя формула показывает, что в объем стоков по счетчику не включаются объемы, вычисленные из объема $V_{сч}$ по дополнительному потреблению, так как эта потребленная вода не сбрасывается в канализацию.

Третий модуль реализует алгоритм начисления объемов по общему водосчетчику для коммунальных квартир и многоквартирных домов. Этот расчет используется с 2011 года в соответствии с нормативным документом [3]. Основная идея данного алгоритма заключается в разделении коммунальной услуги на две составляющие: одна предоставляется внутри квартиры, а вторая относится к общим домовым нуждам (ОДН). Поэтому общий объем потребленного ресурса для абонента без счетчика вычисляется как $V_{абон} = V_{норм} + V_{одн_абон}$, а для абонента со счетчиком $V_{абон} = V_{сч} + V_{одн_абон}$.

Объем $V_{норм} = V_{норм_вод} + V_{норм_сток}$, где $V_{норм}$ – объем услуги по норме. $V_{сч}$ – объем по индивидуальному счетчику, $V_{одн_абон}$ – объем, приходящийся на общедомовые нужды. Объемы $V_{норм_вод}$ и $V_{норм_сток}$ вычисляются по алгоритмам, описанным соответственно в первом и втором модулях, а объем $V_{одн_абон}$, приходящийся на одного абонента (квартиру, комнату) в доме, определяется последующему алгоритму.

1 этап. Вычисляется объем $V_{одн_дом}$ по всему дому:

$$V_{одн_дом} = V_{дом} - V_{норм_дом} - V_{сч_дом},$$

где $V_{дом}$ – объем, вычисленный по показаниям общедомового счетчика; $V_{норм_дом}$ – сумма объемов, начисленных по квартирам в соответствии социальной нормы за месяц; $V_{сч_дом}$ – сумма объемов, начисленных по показаниям индивидуальных счетчиков, установленных в квартирах.

2 этап. Вычисляется объем $V_{одн_абон}$ по каждому абоненту:

а) если объем $V_{одн_абон} > 0$, то $V_{одн_абон} = V_{одн_дом} \times (P_{абон} \div P_{дом})$, где $P_{абон}$ – общая площадь квартиры (комнаты) абонента; $P_{дом}$ – общая площадь всего дома;

б) если объем $V_{\text{одн_абон}} < 0$, то $V_{\text{одн_абон}} = V_{\text{одн_дом}} \times (N_{\text{жс_абон}} \div N_{\text{жс_дом}})$, где $N_{\text{жс_абон}}$ – количество жителей квартиры (комнаты) абонента; $N_{\text{жс_дом}}$ – общее количество жителей всего дома.

Четвертый модуль отвечает за операции учета платежей, поступающих от абонентов. Ввод платежей реализуется разными способами: ручной ввод квитанций контролерами абонентской службы из реестров, поступающих от почты, банковских учреждений и из кассы предприятия; электронная доставка реестровых файлов через Интернет от организаций, с которыми заключены договоры на такого рода услуги. Но в обоих случаях рассматриваемый модуль предварительно накапливает поступающие реестры в базу квитанций, затем по инициативе ответственного работника абонентской службы выполняет автоматическую разноску платежей и показаний счетчиков, а также начисление по этим показаниям объемов и стоимости услуг в финансовые карточки абонентов.

Такое разделение операций ввода квитанции и их разnosки связано со следующими аспектами: разграничение действий персонала и их контроль; защита информации, находящейся в финансовых карточках абонентов; дата поступающих реестров не всегда совпадает с операционным месяцем, а следовательно с датой разnosки платежей; удобство и гибкость при работе с информационной системой. Рассматриваемый функциональный модуль обеспечивает не только автоматический ввод всех данных, необходимых для учета абонентов, но и операции откатов, возврата в исходное состояние до момента разnosки данных. Это требование защищает систему от возможных ошибок пользователей и придает ей устойчивость, живучесть в эксплуатации.

Пятый модуль позволяет персоналу службы после завершения процесса начисления за воду и стоки по всем абонентам выполнить потоковую печать извещений-квитанций по участкам, улицам, домам. При наличии быстродействующего принтера это существенно ускоряет данную операцию.

Все остальные пять модулей выполняют сервисные функции по формированию различных отчетов, справок, документов, файлов, таблиц, а также поддерживают операционные периоды в системе и задачи администрирования. Необходимо отметить, что *восьмой* модуль отвечает за формирование файлов (Excel-формат, DBF-формат), предназначенных для внешних пользователей, которым требуется информация по абонентам (например, дебиторская задолженность, наличие или отсутствие платежей в запрашиваемом месяце).

Изложенные выше проектные решения позволили построить физическую модель абонентской службы в среде системы управления базами данных (СУБД) ClarionforWindows. Данная реляционная модель была применена в разработке программного комплекса “Абонентский отдел (физические лица)” [4]. Для поддержки баз данных была выбрана СУБД Pervasive.SQL, которая является продуктом эволюционного развития СУБД Vtrieve. Эту платформу отличают такие свойства, как высокая производительность, способность к встраиванию, мощность, безопасность, масштабируемость. Отмеченный выше программный комплекс “Абонентский отдел (физические лица)” (рис. 4) был успешно внедрен автором на нескольких водоснабжающих предприятиях Ростовской области и эффективно работает по настоящее время.

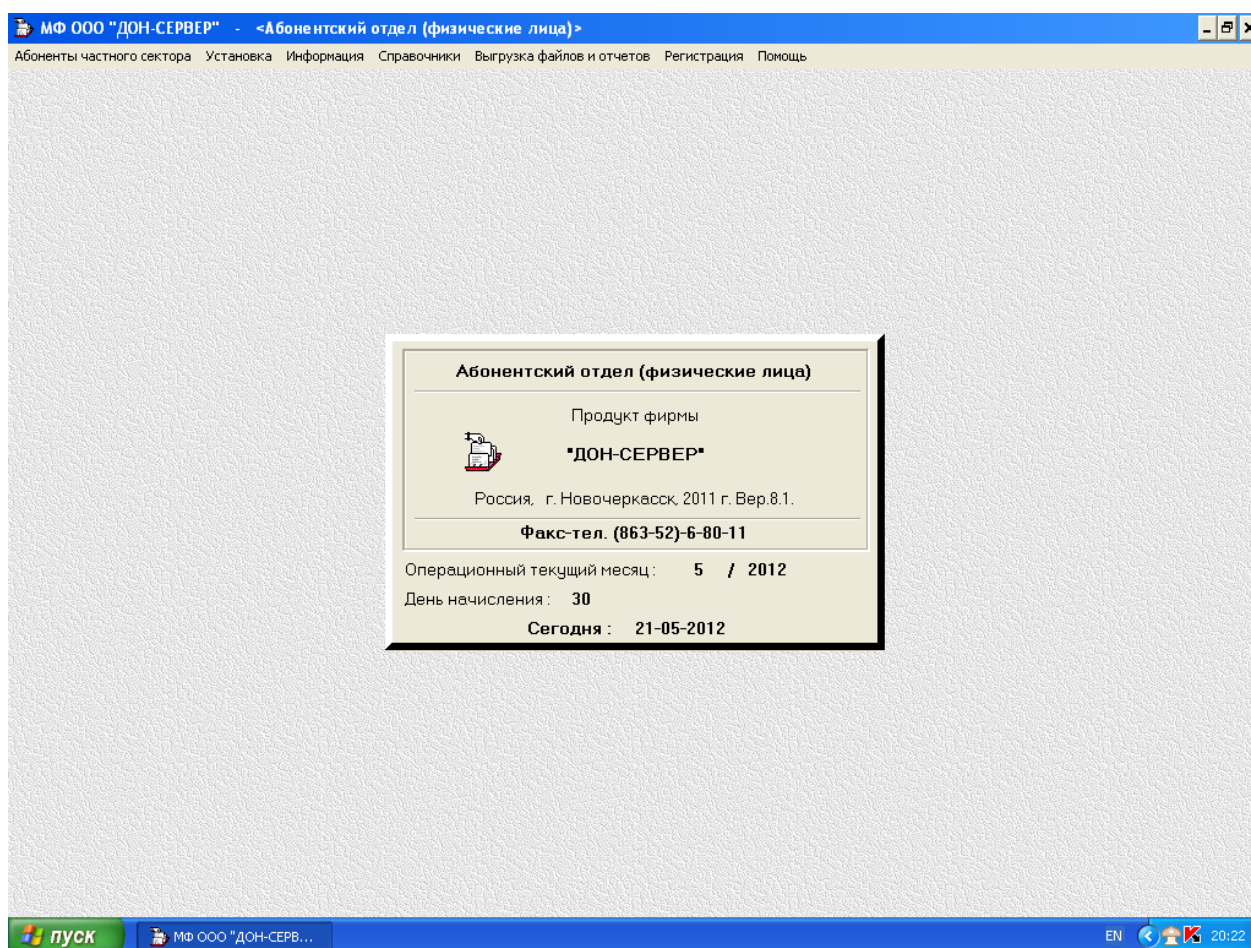


Рис. 4. Главный экран программного комплекса

Литература

1. Кон-фан-сан М.И., Шостак О.И., Нефёдов Б.Н. Биллинговые системы в ЖКХ (Обзор, анализ и принципы проектирования). Материалы VII Всероссийской конференции математическому моделированию и информационным технологиям. – Красноярск, 2006. – С. 88-89.

2. Дейт К. Дж. Введение в системы баз данных = Introduction to Database Systems. – 8-е изд. – М.: «Вильямс», 2006. – 1328 с. – ISBN 0-321-19784-4
3. Постановление Правительства РФ от 06.05.2011 N 354 "О предоставлении коммунальных услуг собственникам и пользователям помещений в многоквартирных домах и жилых домов"-<http://www.consultant.ru/law/hotdocs/13369.html>
4. ООО МФ "Дон-Сервер". Программный комплекс "Абонентский отдел (физические лица)". Документация для пользователя, 2011.

e-mail: don-server@mail.ru

УДК 004.09:338.26.015:323.326

ИНФОРМАЦИОННО-АНАЛИТИЧЕСКАЯ СИСТЕМА ОЦЕНКИ БИЗНЕС-ПЛАНОВ НАЧИНАЮЩИХ ПРЕДПРИНИМАТЕЛЕЙ

Костеникова Ю.С., Щербакова Е.А.

Южно-Российский государственный технический университет (НПИ), г. Новочеркасск

Описана информационно-аналитическая система оценки бизнес-планов начинающих предпринимателей, стоящих на учете в центре занятости. Приведены возможности и преимущества использования для предприятия с позиции эффективности и экономии. Рассмотрены показатели оценки эффективности бизнес-проекта.

В целях обеспечения потребностей города Миллерово в расширении производства товаров и услуг посредством самозанятости населения Государственное казенное учреждение Ростовской области "Центр занятости населения города Миллерово" оказывает содействие безработным гражданам.

Государственная программа «Помощь самозанятости населения» состоит из годового пособия гражданина, признанного безработным, выданного единовременно, а не выплачиваемого ежемесячно в течение года. Для того чтобы получить финансовую помощь по данной программе, необходимо встать на учет в Центр занятости населения в общем порядке. В течение двух месяцев гражданин, вставший на учет, находится в поисках работы и, в случае, если подходящая работа не была найдена, по истечении этого срока человек признается безработным.

Являясь безработным, соискатель имеет право написать заявление о предоставлении финансовой помощи для малого бизнеса, предоставив бизнес-план. Бизнес-план отправляется на экспертизу в стороннюю организацию. В случае положительного заключения экспертов центр занятости

подписывает с гражданином договор о предоставлении финансовой поддержки и перечисляет средства на его расчетный счет в банке.

Схема получения субсидии представлена на рис. 1.

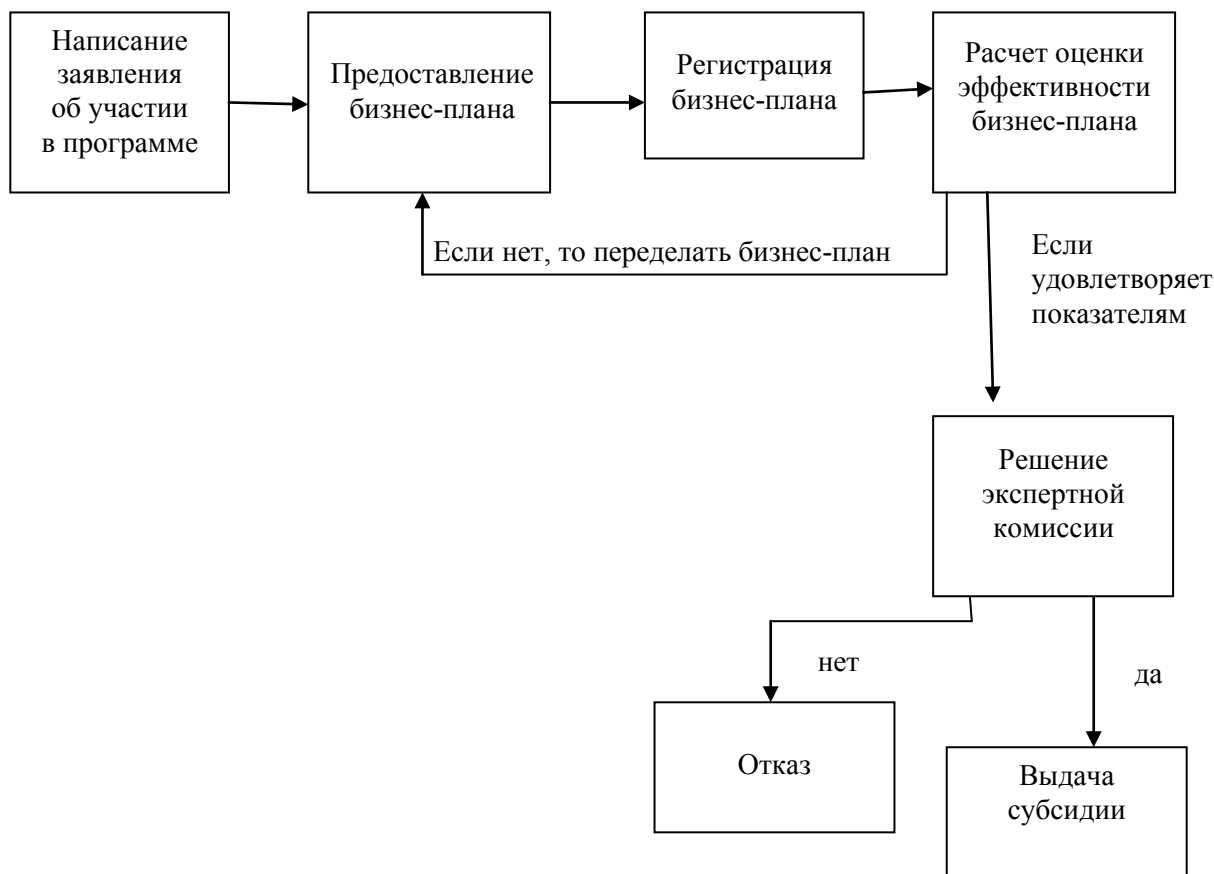


Рис. 1. Схема получения субсидии

Бизнес-план составляется в упрощенном варианте. Для оценки его эффективности необходимо рассчитать следующие показатели:

- а) чистый дисконтированный доход (NPV).
- б) индекс доходности (PI).
- в) срок окупаемости (PP).
- г) внутренняя норма доходности (IRR).

Чистый дисконтированный доход (NPV) позволяет получить наиболее обобщенную характеристику результата инвестирования, т.е. его конечный эффект в абсолютной сумме. Расчет чистого дисконтированного дохода осуществляется по формуле (1):

$$NPV = \sum_{t=t_H}^T (R_t - Z_t) \cdot \alpha_t.$$

Если $NPV < 0$, то в случае принятия проекта инвесторы получают убытки; если $NPV = 0$, то в случае принятия проекта благосостояние инве-

сторгов не изменится, но объемы производства возрастут; если $NPV > 0$, то в случае принятия проекта инвесторы получают прибыль.

Индекс доходности (PI) характеризует доход на единицу затрат и рассчитывается по формуле (1):

$$PI = \frac{1}{k} \cdot \sum_{t=t_H}^T (R_t - Z_t^P) \cdot \alpha_t.$$

Если $PI \leq 1$, то проект должен быть отвергнут в связи с тем, что он не принесет прибыль; если $PI > 1$, то в случае принятия проекта инвесторы получают прибыль.

Срок окупаемости (PP) показывает число базовых периодов, за которое исходная инвестиция будет полностью возмещена за счет генерируемых проектом притоков денежных средств. PP рассчитывается по формуле (1):

$$PP = \frac{\text{Инвестиции}}{\text{Чистая прибыль}}$$

Внутренняя норма доходности (IRR) представляет собой ту норму дисконта, при которой величина приведенных эффектов равна приведенным капиталовложениям и рассчитывается по формуле (1):

$$IRR = E_1 + \frac{NPV(E_1)}{NPV(E_1) - NPV(E_2)} \cdot (E_2 - E_1)$$

Процедура оценки эффективности бизнес-планов проходила вручную и поэтому занимала длительное время. В связи с этим была предложена информационно-аналитическая система оценки эффективности бизнес-планов государственного казенного учреждения Ростовской области "Центр занятости населения города Миллерово", которая позволит значительно сократить время обработки информации и повысить эффективность работы специалистов.

Существует множество корпоративных информационных систем, в которых реализованы решения данных задач. К таким системам относятся: ИНЭК-Холдинг, Альт-Инвест, Project Expert, Business Plan PL Professional category. Однако данные системы слишком дороги и не подходят для малых предприятий. В связи с этим было принято решение создать альтернативную систему на платформе 1С: Предприятие 8.2. Программное обеспечение разрабатывалось как отдельный модуль.

Предлагаемая информационно-аналитическая система оценки бизнес-планов начинающих предпринимателей в рамках государственной программы "Помощь самозанятости населения" позволяет:

- автоматизировать оценку эффективности бизнес-планов с целью сокращения времени на принятие решений по выдаче субсидии;

- сформировать единую информационную базу данных о безработных, желающих принять участие в данной программе, собрать общие сведения об участниках программы;
- сформировать единую информационную базу данных о бизнес-планах, принимающих участие в данной программе;
- создать систему отчетности;
- вести статистику выдачи субсидий.

Литература

1. Савельева Н.А., Бринк И.Ю. Бизнес-план предприятия. Теория и практика. – Р н/Д: Феникс, 2007. – 384 с.

УДК 004.9

ГРАФИЧЕСКАЯ БИБЛИОТЕКА В СОСТАВЕ АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ ПЛАНИРОВАНИЯ РС МКС

Орловский Н.М.

Южно-Российский государственный технический университет (НПИ), г. Новочеркасск

Перечислены основные отличия и новые возможности графической библиотеки в составе АСП РС МКС по сравнению с существующим комплексом программ, используемым при планировании полетов для Российского сегмента МКС.

В предыдущих работах [1,2] говорилось о создании, отличительных особенностях и возможностях новой автоматизированной системы планирования. Традиционно в центре управления полетами в существующем комплексе программных инструментов для разработки планов используется графическое представление, так как информация в этом виде наиболее удобна для восприятия. Естественно, что и в новой системе наличие графического формата является обязательным условием. Принципиальным отличием автоматизированной системы планирования можно назвать такую возможность, как представление любого уровня плана в двух форматах одновременно: табличном и графическом (рис. 1).

Причем в новой системе источником данных для обоих форматов служит единая база данных полетных операций и планов, в которой хранятся установочные параметры атрибутов для каждой полетной операции. При этом обеспечивается синхронизация данных между форматами.

При планировании полетов используется метод «от общего к частному», то есть поэтапное увеличение детализации иерархической структуры плана на интервале всей экспедиции, далее на интервале недели и на интервале планируемых суток [3].

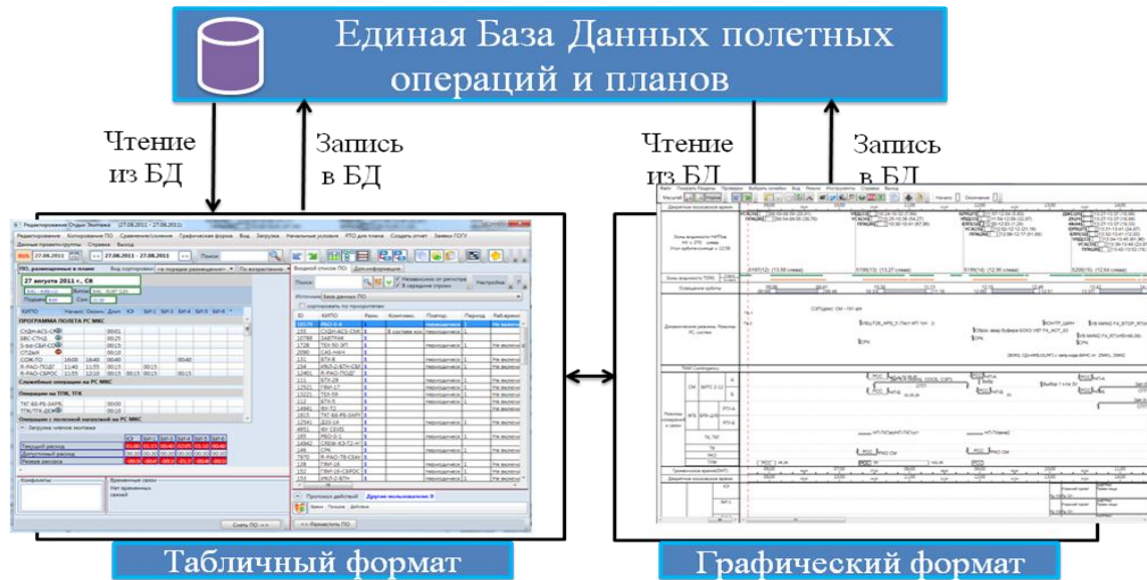


Рис. 1. Форматы представления планов в АСП РС МКС

Графическая библиотека (оболочка) представляет собой универсальный конструктор для формирования определенной структуры плана в зависимости от его типа и назначения. Планы уровня ДПП, БДПП и ДПП ТК/ТКГ непосредственно используются только в графическом виде, и их создание производится в графическом формате (рис. 2).

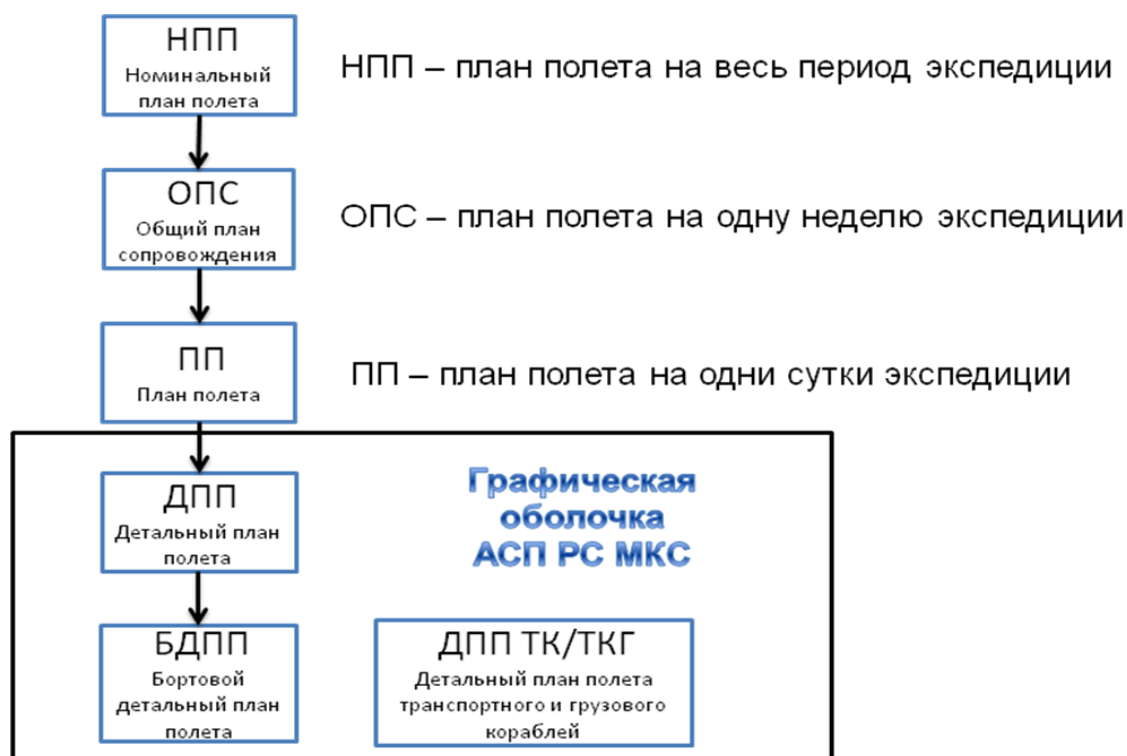


Рис. 2. Редактор планов АСП РС МКС

Модуль детального плана полета представляет собой графическое отображение полетных операций во времени. Детальный план полета отражает привязку к определенному моменту времени выполняемых экипажем служебных и целевых полетных операций и процедур, в том числе, связанных с обеспечением нормального режима труда и отдыха членов экипажа, а также выполнением автоматических динамических режимов, режимов измерения и связи.

Модуль разработки и работы с бортовым детальным планом полета строится на основе ДПП и содержит лишь ту информацию, которой достаточно для выполнения полета на борту. БДПП для космонавта является инструкцией к действию.

Специальный графический формат ДПП ТК/ТКГ создается для детального планирования полетов транспортных и грузовых кораблей, доставляющих космонавтов и грузы к станции.

Графическая оболочка автоматизированной системы планирования РС МКС по своей сути является редактором, в котором оператор-планировщик создает план с помощью специальных инструментов. У всех трех графических форматов, описанных выше, имеются как универсальные инструменты для работы, так и строго характерные для конкретного типа плана.

При разработке графической оболочки был сделан инструмент, который позволяет формировать графические элементы различного уровня

сложности и вложенности, в том числе такие элементы, как «ориентация», «циклограмма», «программа», «НП+В», «Запись» и др. Структура каждого элемента хранится в xml файле, который обрабатывается при запуске графических форматов (рис. 3).

Отличительной особенностью новой системы планирования является разработка алгоритмов автоматического динамического заполнения некоторых разделов данными, которые формируются на основе анализа и обработки остальных разделов плана. Заполнения разделов происходит по специальным алгоритмам и в определенной очередности. Это существенно уменьшает время, требуемое на создание готового плана.

Результат каждого действия планировщика на любой редактируемый элемент плана автоматически заносится в базу данных. При ошибочной манипуляции можно воспользоваться инструментом «Отмена последнего действия». Можно задать и зафиксировать собственный стиль отображения информации (задание состава и порядка расположения разделов, их размеры и другие настройки), что делают труд планировщиков более удобным и быстрым.

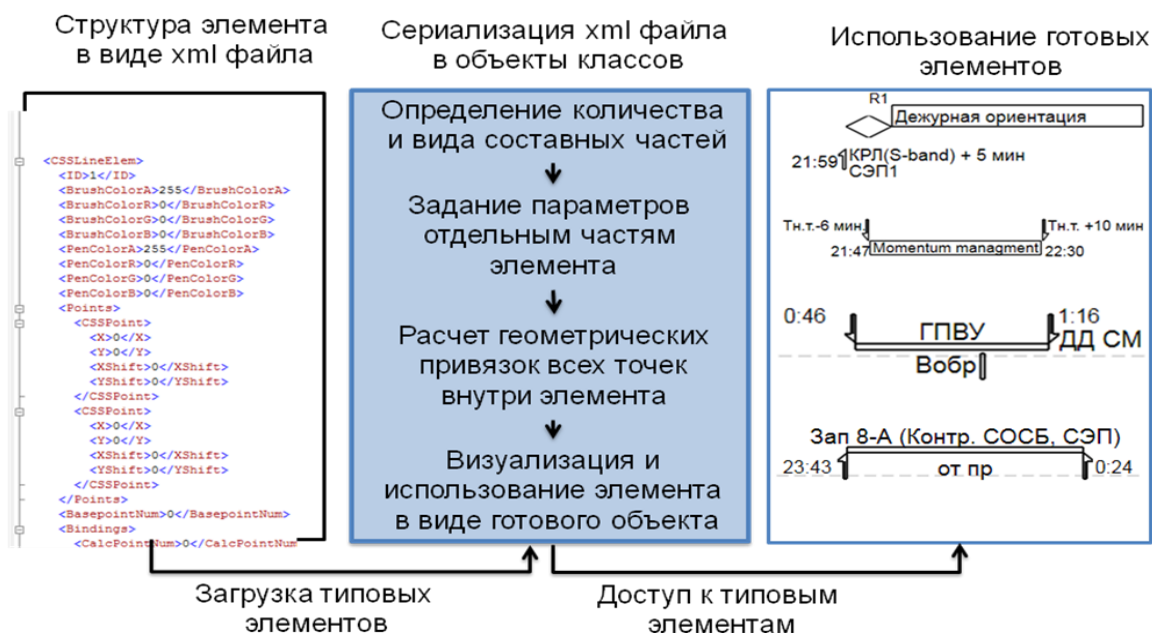


Рис. 3. Создание типовых элементов

В новой системе планирования уровень автоматизации избавляет планировщика от рутинного процесса подсчета приоритетов и поиска мест для размещения полетной операции. Система сама выполнит упорядочивание по приоритетам и предложит оператору готовые варианты размещения полетной операции (рис. 4).

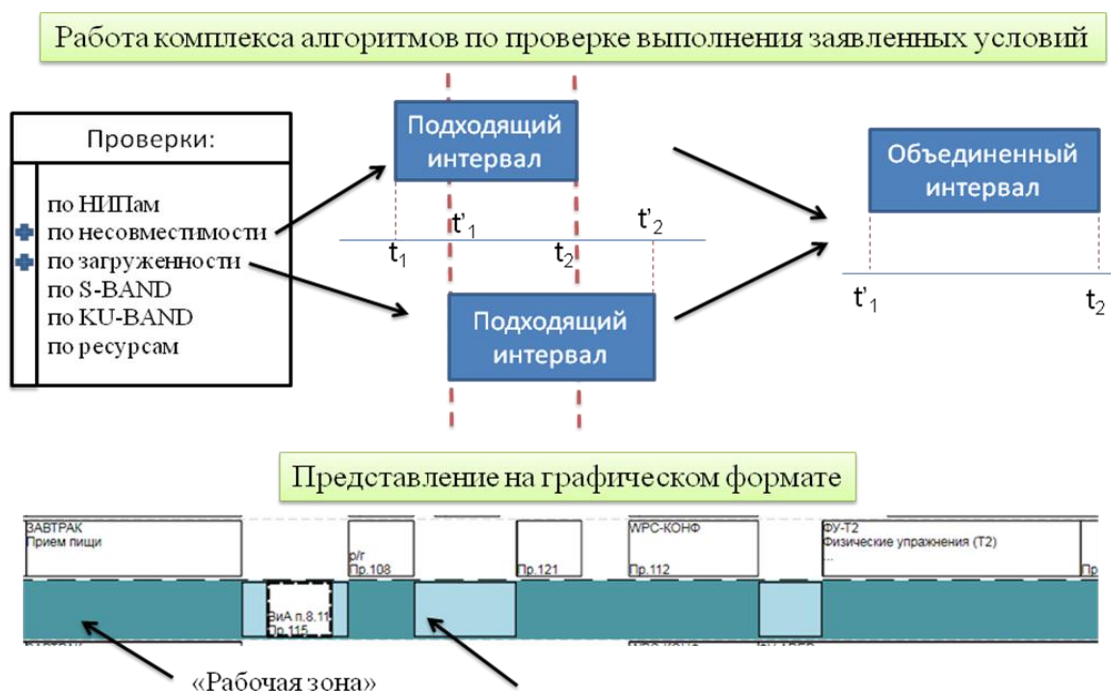


Рис. 4. Автоматическая проверка заявленных условий

Разработанные алгоритмы проверяют сочетание выполнимости требований на загруженность члена экипажа, несовместимость полетных операций, наличия ресурсов, зон S-band и Ku-band, зон НИПов. При смещении полетной операции в так называемую «рабочую зону» планировщик видит, где можно ее разместить, потому как система подсвечивает прямоугольными областями места, удовлетворяющие выбранным проверкам. Остается поставить полетную операцию в указанную позицию.

Дальнейшее перспективное направление – увеличение уровня автоматизации новой системы планирования и создания алгоритмов оптимизации размещения полетных операций. Это позволит в короткое время формировать версию допустимого плана, и это особенно важно, когда нужно произвести перепланирование при возникновении отказов и других нештатных ситуаций.

Литературы

1. Орловский Н. М. Автоматизированная система планирования российского сегмента международной космической станции // Теория, методы проектирования, программно-технологическая платформа корпоративных информационных систем: материалы VIII Междунар. науч.-практ. конф., г. Новочеркасск, 2010г. / Юж.-Рос. гос. тех. ун-т (НПИ). – Новочеркасск: ЮРГТУ, 2010. – С. 35-40.
2. Орловский Н. М. Обзор Модуль детального плана полетов автоматизированной системы планирования российского сегмента международной космической станции // Теория, методы проектирования, программно-технологическая платформа корпоративных информационных систем: материалы VIII Междунар. науч.-практ. конф., г. Новочеркасск, 2010г. / Юж.-Рос. гос. тех. ун-т (НПИ). – Новочеркасск: ЮРГТУ, 2010. – С. 41-44.

3. Станиловская В. И. Автоматизация планирования полетов долговременных орбитальных комплексов: дис., канд. тех. наук, Королев, 2008.

УДК 004

ВВЕДЕНИЕ В ПРОЦЕСС ПЛАНИРОВАНИЯ ПОЛЕТОВ РС МКС

Орловский Н.М.

Южно-Российский государственный технический университет (НПИ), г. Новочеркасск

Рассмотрены основные уровни и этапы процесса планирования РС МКС. Описан процесс и принципы построения планов на стратегическом, тактическом и исполнительном уровнях. Проанализирован перечень объектов и их параметров, которые являются входными данными на этапе планирования действий экипажа РС МКС.

Планирование представляет собой вид деятельности, связанный с постановкой целей и действий в будущем, направленных на реализацию этих целей. Планирование в самом общем виде подразумевает выполнение следующих этапов:

- постановка целей и задач;
- составление программы действий;
- выявление необходимых ресурсов и их источников;
- определение непосредственных исполнителей и доведение планов до них.

Планирование стремится к наиболее оптимальному распределению ресурсов для достижения поставленной цели.

При изучении особенностей организации процесса планирования полета орбитального комплекса (ОК) целесообразно четко понять весь механизм формирования планов и их исполнения. Для этого следует сказать о трехуровневой организации процесса планирования: на макроуровне, на мезауровне (собственном уровне системы) и на микроуровне [1].

Так, на собственном уровне организационной системы планирования ее исследование сводится к функционально-структурному анализу групп планирования (ГП). На макроуровне уже необходимо учитывать функциональную структуру всей Главной Оперативной Группы Управления (ГОГУ), обеспечивающей комплексное управление полетом. А на микроуровне может рассматриваться коллективная и индивидуальная деятельность специалистов планировщиков различных категорий.

Нас интересует представление процесса планирования на микроуровне. Иерархия уровней организации планирования отражает иерархию

задач планирования, которые связаны с построением семейства планов различной степени полноты и срочности. Поэтому для процесса планирования полета ОК характерно следующее [2]:

- степень детализации при разработке плана полета ОК должна соответствовать уровню планирования, а язык описания (моделирования) – цели исследования и характеру информации, располагаемой планировщиком;
- для каждого уровня планирования существует некоторая своя область неопределенностей значений параметров и критериев планирования. Эта неопределенность является не случайным, а систематическим фактором: она обусловлена объективным содержанием задачи планирования, особенности восприятия, мышления и языка специалистов.

Планирование полета заключается в конечном счете в установлении состава и временной последовательности операций, приводящей к достижению целей полета, а также в определении порядка координированной работы всех служб и средств системы управления в ходе выполнения операций [3].

Разработка плана, начинаемая еще на этапе проектирования, представляет собой основу процесса управления полетом. Управление полетом должно осуществляться в строгом соответствии с принятым к исполнению планом.

Для формирования исходного и оперативного планов применяют одни и те же методы. Однако существенным отличием оперативного планирования от дополётного является ограниченное время его выполнения, иногда и дефицит времени.

План полета разрабатывается в нескольких видах, соответствующих определенному уровню.

На высшем, стратегическом, уровне планирования формулируются цели полета, устанавливаются средства и способы их достижения, необходимые ресурсы и принципы их использования, а также основные ограничения, которые нужно учитывать при разработке плана. Документы этого уровня являются основой для разработки планов следующего уровня и могут уточняться в ходе полета в зависимости от реальной полетной обстановки. Подобные документы представляют собой основные положения по программе полета и являются, по существу, заданием на полет.

Следующий уровень планирования – тактический. Он определяет последовательность и методы выполнения операций, приводящих к достижению цели полета, порядок использования технических средств в составе системы управления полетом, программу обмена информацией меж-

ду звеньями контура управления полетом. Документами этого уровня являются номинальный, долгосрочный и краткосрочный планы полета.

Номинальный план полета (НПП) охватывает либо весь полет, либо большую его часть и определяет последовательность основных операций к выполнению на выбранном интервале планирования. В процессе выполнения полета НПП периодически корректируется с учетом обстановки. Такой оперативно измененный номинальный план называется долгосрочным планом полета.

Краткосрочный план полета, называемый также оперативным планом сполета (ОПП), основан на данных номинального или долгосрочного плана, реальной полетной обстановке и фактическом состоянии КА в текущий момент времени. Он определяет более детализированный состав, последовательность и время проведения полетных операций, необходимых в ближайший период, равный одной – двум неделям. При разбивании ОПП по суткам получают план полета (ПП).

Затем следует исполнительный уровень, на котором разрабатывается детальный план полета (ДПП) – наиболее подробный руководящий документ. ДПП формируется на каждые сутки полета на основе краткосрочного плана с методиками выполнения планируемых операций, оперативных данных о реальной полетной обстановке и текущей баллистической информации. Также в качестве исходной информации используются материалы:

- подробные описания операций, включенных в состав работ на рассматриваемые сутки полета;
- методики управления бортовыми системами КА;
- при оперативном (в ходе полета) формировании детального плана – фактические данные о текущем состоянии бортовых систем;
- данные о состоянии технических средств, входящих в систему управления полетом;
- баллистическую информацию на планируемые сутки полета;

К плановым документам исполнительного уровня также относят инструкции по выполнению отдельных операций.

Ввиду указанных особенностей процесс планирования полета современных КА в настоящее время не представляется возможным полностью формализовать для автоматического выполнения. Возможна автоматическая разработка только некоторых компонентов плана. Поэтому планирование полета проводится подготовленным персоналом с помощью

вспомогательных технических средств и программного обеспечения, образующих систему автоматизированного планирования полета [3].

При разработке планов полета тактического уровня окончательная оценка качества плана, разрабатываемого наземным персоналом или экипажем, с целью выбора наиболее приемлемого варианта, осуществляется экспертным путем. Эксперты учитывают такие требования, как стремление к максимальной степени достижения целей полета при заданном уровне безопасности экипажа, обеспеченность плана полета ресурсами, выполнимость плана, гибкость, т.е. возможность оперативной трансформации плана с соблюдением установленных правил и ограничений при возникновении аномальных ситуаций.

Вследствие взаимосвязанности всех компонентов детального плана к их разработке предъявляют разнообразные, иногда противоречивые требования, поэтому формирование ДПП происходит в несколько приближений [3].

При разработке планов оператор-планировщик ГП должен соблюдать требования, приведенные ниже.

В первую очередь планируют служебные операции, требующие привязки к определенным суточным виткам или моменту времени. Следующими после служебных планируют целевые операции, имеющие баллистические ограничения по времени выполнения (целевые операции первого типа), а затем – целевые полетные операции, не имеющие баллистических ограничений (целевые операции второго типа). Для операций, которые необходимо провести именно в данные сутки полета, желательно предусмотреть резерв времени в течении этих суток на повторение работ в случае невыполнения их в запланированное время.

При планировании за основную единицу плана полета принимается полетная операция – логически и технологически завершенная работа, направленная на получение полезного результата (требуемое изменение состояния комплекса). По функциональному назначению полетные операции можно разделить на следующие категории – служебные, бытовые и целевые [1].

Каждая полетная операция расходует известное количество определенных ресурсов. К числу располагаемых ресурсов ОК относятся имеющиеся запасы:

- топлива;
- электроэнергии;
- кислорода;

- воды и пищи для экипажа;
- видео, фотопленок и других носителей информации;
- допустимая продолжительность работы экипажа;
- возможность связи ОК с наземным комплексом управления;
- количество членов экипажа.

Одним из критериев выполнимости совокупности полетных операций $P = \{p_i, i = \overline{1, m}\}$, планируемых к реализации на интервал времени $[t_1, t_2]$, является критерий достаточного количества ресурсов:

$$\sum_{i=1}^m Q_{ji} \text{potr} \leq Q_j \text{raspol}, \quad (1)$$

где $Q_{ji} \text{potr}$ – количество j -ого вида ресурсов ОК, потребляемое i -ой полетной операцией. $Q_j \text{raspol}$ – располагаемое количество j -ого вида ресурсов ОК на планируемом интервале времени.

Как правило, при решении оптимизационных задач критерий расхода ресурсов всегда сводится к минимуму. В данной области задача использования всего запаса некоторых ресурсов на заданном интервале приветствуется в силу того, что их время существования ограничено указанным интервалом. На момент окончания интервала остаток этих ресурсов будет обнулен и перерасчитан на следующие сутки. К таким ресурсам относятся: полетное и рабочее время, энергоресурсы на сутки, средства НКУ в пределах суток и т.д.

На каждые сутки полета формируется список ПО, которые нужно реализовать экипажу РС МКС. Если сформированная совокупность ПО не удовлетворяет ограничениям, то задача выбора из списка этих операций решается с использованием их приоритетов, устанавливаемых в соответствии с определенными правилами. Главными параметрами операций, учитываемыми при установке приоритета, являются их важность и срочность.

Важность операции определяется:

- ролью отдельной операции (ее результата) на каждом конкретном этапе полета, то есть зависимостью общего положительного итога программы от выполнения данной операции;
- взаимной зависимостью задач, то есть зависимостью успешности решения одной частной задачи от результатов другой в рамках общего процесса достижения цели полета.

Срочность операции определяется:

- недолговечностью или непостоянством во времени некоторых ресурсов, необходимых для выполнения данной планируемой операции;
- вероятность значимых событий, к которым относятся жизненно важные для программы полета операции;
- возможность перемещения операции в пределах рабочего дня экипажа в зависимости от ограниченности некоторых видов ресурсов на определенном интервале.

Приоритеты ПО делятся на фиксированные и переменные. Фиксированный приоритет ПО задается на весь полет. Переменный приоритет ПО зависит от текущих условий формирования плана и описывается нечеткой функцией:

$$pr = pr(N_{pr}, f_{pr}, t_{pr}, x_{pr}, R_{pr}, R_{pr}), \quad (2)$$

где:

1) N_{pr} – оценка количества включенных в план операций различных типов. Под типом операции следует понимать ее принадлежность одной из следующих групп: стыковка, расстыковка, перестыковка; внекорабельная деятельность; медицинский контроль экипажа; материально-техническое обеспечение; проведение научных экспериментов; работа со средствами массовой информации. Операции, связанные со стыковкой, наиболее важны и имеют наивысший приоритет среди остальных групп.

2) f_{pr} – оценка частоты условий проведения операции. Чем чаще совокупность условий проведения операции встречается на определенном промежутке времени, тем приоритет полетной операции ниже. По ходу наполнения плана количество условий ее проведения сокращается, тем самым растет необходимость реализовать ПО в подходящий момент.

3) t_{pr} – оценка наступления назначенного времени проведения операции. Полетная операция может иметь широкий интервал времени, внутри которого она может быть выполнена. В то же время, ей задан точный подинтервал, на котором результативность ее исполнения будет максимальной. При попадании операции в такой промежуток приоритет резко возрастает, что делает ее лидером в списке на размещение.

4) x_{pr} – оценка достижения «географических» координат, необходимых для проведения операции. Здесь имеется в виду достижение космическим аппаратом определенных параметров орбиты по отношению к космическим объектам. При их совпадении с заявленными условиями проведения полетной операции приоритет последней увеличивается.

5) R_{pr} – оценка ресурса экипажа, требуемого для операции. На начальном этапе распределения ресурсы предоставлены в полном объеме для любой из операций. Но по мере приближения к концу формирования плана их запас резко уменьшается. В связи с этим понижается вероятность выполнения работы, и тогда ее приоритет возрастает.

6) R_{pr}' – оценка энергообеспеченности операции. Здесь энергообеспеченность аналогична другим ресурсам, за исключением некоторого отличия – на станции есть резервные источники, которые кратковременно могут восполнить чрезмерное потребление энергии. Поэтому требуется расчет возможности выполнения энергоемкой операции внутри окна восполнения нехватки энергии.

Кроме приоритета у ПО есть понятие ее совместимости с выбранными ранее и включенными в план ПО. Совместимость S операций ПО1 и ПО2 определяется условиями их одновременного выполнения или следования одной за другой, а также использования одних и тех же обобщенных ресурсов (аппаратуры, канала связи и т.д.).

Каждой ПО из списка на распределение задан коэффициент эффективности по всем членам экипажа. Эффективность означает выбор наилучшего варианта для выполнения ПО из имеющихся возможностей и выполнение его наилучшим образом. Значение коэффициента эффективности находится в диапазоне $[0,1]$. Космонавт максимально эффективно выполняет ПО при показателе, равном 1, и не ставится оператором ГП для реализации операции, если его личный коэффициент для этой ПО равен 0.

Таким образом, перед началом формирования плана функционирования экипажа РС МКС заданы:

- список ПО, которые необходимо разместить в плане – поступает из НПП и программы полета;
- список ресурсов и их доступное количество на заданный интервал – поступают из программы полета;
- приоритеты ПО, которые определяют их положение на интервале – фиксированные приоритеты рассчитаны заранее и известны на момент формирования действий экипажа, а переменные приоритеты рассчитываются в процессе построения ПП;
- ресурсы каждой ПО, которые нужно использовать для ее исполнения – поступают из программы полета и полетных процедур;
- список несовместимых ПО по времени и ресурсам – поступают из программы полета и полетных процедур;

- коэффициенты эффективности для каждого члена экипажа по всем ПО из списка – поступают из программы полета;
- совокупность ограничений и правил для проведения полета – корректируются на протяжении формирования всех уровней планов до момента создания ДПП с учетом приобретенного опыта и фактического состояния КА и НКУ.

Зная все эти данные, необходимо составить структуру формализованной модели функционирования экипажа РС МКС и на ее основе сформировать критерии оптимального построения последовательности и набора полетных операций из заявленного списка с учетом следующих позиций:

- сформированная последовательность должна удовлетворять ограничениям и условиям проведения полета;
- по возможности включить в набор как можно больше ПО, имеющих высокий коэффициент эффективности по членам экипажа;
- если не все ПО возможно разместить в наборе, то отбросить наименее приоритетные;
- ПО с переменным приоритетом по возможности выполнялись в зонах, возводящих их приоритет в максимальное значение;
- остаток ресурсов, обнуляемых по завершению суток, попытаться минимизировать, а остаток остальных ресурсов – максимизировать.

Дальнейшей задачей является разработка формализованной математической модели для оптимизации вышеуказанных задач. Учитывая сложность и разнообразность подцелей планирования критерий оптимизации должен быть составным с использованием весовых коэффициентов, определяемых на основе экспертных оценок.

Литература

1. Станиловская В. И. Автоматизация планирования полетов долговременных орбитальных комплексов: дис., канд. тех. наук, Королев, 2008. – 198 с.
2. Аверкин А. Н., Батыршин И. З., Блишун А. Ф. Нечеткие множества в моделях управления и искусственного интеллекта. – М.: Наука, 1986. – 312 с.
3. Соловьев В.А., Лысенко Л.Н., Любинский В.Е. Управление космическими полетами. Часть I. – М.: Изд-во МГТУ им. Н. Э. Баумана, 2009. – 476 с.

КОНЦЕПЦИЯ ПОСТРОЕНИЯ СИСТЕМЫ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ НА ОСНОВЕ ПОНЯТИЯ НЕЧЕТКОЙ ОНТОЛОГИИ

Погорелов А.С.

Южно-Российский государственный технический университет (НПИ), г. Новочеркасск

Рассматривается возможная концепция построения системы поддержки принятия решений на основе понятия нечеткой онтологии. Вводится ряд определений, касающихся онтологии и теории нечетких множеств. Предметная область для рассматриваемой системы – это процессы парирования непредусмотренных ситуаций, возникающих на борту МКС при осуществлении экипажем штатной деятельности. В упрощенном виде представляется структура нечеткой онтологии.

Система поддержки принятия решений (СППР) (англ. Decision Support System, DSS) – компьютерная автоматизированная система, целью которой является помощь людям, принимающим решение в сложных условиях для полного и объективного анализа предметной деятельности. Для анализа и выработок предложений в СППР используются разные методы. Это могут быть: информационный поиск, интеллектуальный анализ данных, поиск знаний в базах данных, рассуждение на основе прецедентов, имитационное моделирование, эволюционные вычисления и генетические алгоритмы, нейронные сети, ситуационный анализ, когнитивное моделирование и др [1].

Перечисленные выше методы отражают непосредственно способ принятия решений или выбора наилучшего варианта из множества альтернатив. Однако любые методы должны быть органично соединены в систему, которая предназначена для информационной поддержки процесса решения задач в какой-либо предметной области. Для создания общего представления о СППР и принципах ее функционирования необходимо описать концепцию ее построения. Концепция построения СППР предполагает ответы на следующие основные вопросы:

- 1) Какова структура системы? Какие основные элементы (подсистемы) входят в ее состав?
- 2) Каковы основные функции элементов системы?
- 3) Какие методы используются для реализации поддержки принятия решений? Как эти методы взаимосвязаны между собой?
- 4) Как происходит взаимодействие элементов системы между собой? Что является результатом этого взаимодействия?

В настоящей статье рассматривается понятие нечеткой онтологии как основы для построения системы поддержки принятия решений. Прежде всего, необходимо дать определение онтологии.

Онтология – это точная спецификация концептуализации [2]. Под концептуализацией понимается абстрактная модель явлений (процессов) в мире, составленная посредством определения существенных для описания данных явлений понятий. Точность подразумевает, что типы используемых понятий и ограничения на область применения данных понятий явно определены. Данное определение, введенное Т. Gruber, считается общепринятым определением онтологии [3].

Онтология предназначена для описания знаний, используемых в СППР для выработки рекомендаций. При этом знания в онтологии делятся на два вида: декларативные и процедурные.

Декларативные знания – это знания об объектах предметной области, их структуре, свойствах, ограничениях и т. д. Говоря простыми словами, декларативная база знаний – это обычная база данных, построенная в соответствии с известной моделью данных (иерархическая, сетевая, реляционная).

Процедурные знания – это знания о том, как используются декларативные знания. Например, в качестве процедурных знаний могут выступать алгоритмы действий персонала при возникновении нештатных ситуаций на исследуемом объекте. При этом алгоритм не является линейным, а выбор шагов зависит от информации, хранящейся в декларативной базе знаний.

В настоящей статье используется понятие нечеткой онтологии, поэтому целесообразно ввести некоторые определения из теории нечетких множеств, которые будут необходимы в дальнейшем.

Нечеткое множество – это подмножество A элементов некоторого универсального множества E такое, что каждому элементу сопоставлена степень принадлежности этого элемента данному множеству A . Нечеткое множество полностью определяется заданием функции принадлежности $\mu_A(x)$, область определения которой принадлежит универсальному множеству, а область значений попадает под единичный интервал $[0, 1]$. Чем выше значение $\mu_A(x)$, тем выше оценивается степень принадлежности элемента $x \in \oplus$ нечеткому множеству A .

Именованное нечеткое множество называется нечеткой переменной.

Лингвистическая переменная (в упрощенном смысле) – это переменная, принимающая значения из множества слов или словосочетаний

некоторого естественного языка. Формально, лингвистическая переменная – это набор $\langle b, T, X, G, M \rangle$, где:

b – наименование лингвистической переменной;

X – область определения переменной;

T – множество ее значений, называемых термами (базовое термножество), представляющих собой наименования нечетких переменных на области определения X ;

G – синтаксическая процедура, позволяющая генерировать новые термы.

M – семантическая процедура, превращающая каждый терм из $G(T)$ в нечеткую переменную.

Выделив основные определения, касающиеся онтологии и теории нечетких множеств, необходимо отметить, что на настоящий момент в литературе нет единого мнения о том, как нужно строить онтологию, нет универсальных алгоритмов построения онтологии. Например, в работе [3] выделяются лишь критерии для проектирования онтологии, такие как понятность, согласованность, расширяемость и т.д. В то же время некоторыми авторами осуществлялись попытки стандартизировать процесс построения онтологии [4, 5].

Анализ литературы позволяет сделать вывод, что правила построения, структура, ограничения онтологии для конкретной предметной области должны определяться непосредственно разработчиком. При этом разработчик должен руководствоваться известными на настоящий момент и проверенными на практике принципами, критериями для проектирования.

В соответствии с описанными выше теоретическими аспектами предлагается концепция построения системы поддержки принятия решений на основе понятия нечеткой онтологии. Предметная область для рассматриваемой СППР – это процессы парирования непредусмотренных ситуаций, возникающих на борту МКС при осуществлении экипажем штатной деятельности.

Основой рассматриваемой СППР является нечеткая онтология, состоящая из двух частей:

1) Декларативная база знаний, содержащая знания о предметной области: типы аварийных ситуаций, средства парирования, конфигурация МКС и т. д. При этом знания могут представляться, например, в соответствии с объектно-ориентированным принципом. Данный принцип кажется наиболее удобным для программной реализации нечеткой онтологии.

2) Процедурная база знаний, содержащая перечень алгоритмов выхода из нештатных ситуаций, представленных в виде блок-схем.

Как декларативная, так и процедурная составляющие онтологии содержат в себе элементы нечеткости.

В декларативной базе знаний нечеткость представляется в виде лингвистических переменных, с помощью которых происходит описание данных в онтологии.

Нечеткость процедурной базы знаний означает, что переход от одного узла алгоритма к другому осуществляется на основе нечеткого логического вывода. Реализация нечеткого логического вывода позволяет на каждом шаге принятия решений выбирать наиболее подходящую альтернативу исходя из фактической актуальной информации, хранящейся в декларативной базе знаний.

Еще одной частью нечеткой онтологии является специально разработанный алфавит, который служит для формирования и формализации декларативной и процедурной баз знаний и обеспечения взаимодействия и обмена информацией между частями нечеткой онтологии. Алфавит нечеткой онтологии может содержать определения нечетких термов, модификаторов, кванторов и т. д.

Упрощенная структура нечеткой онтологии представлена на рис. 1.

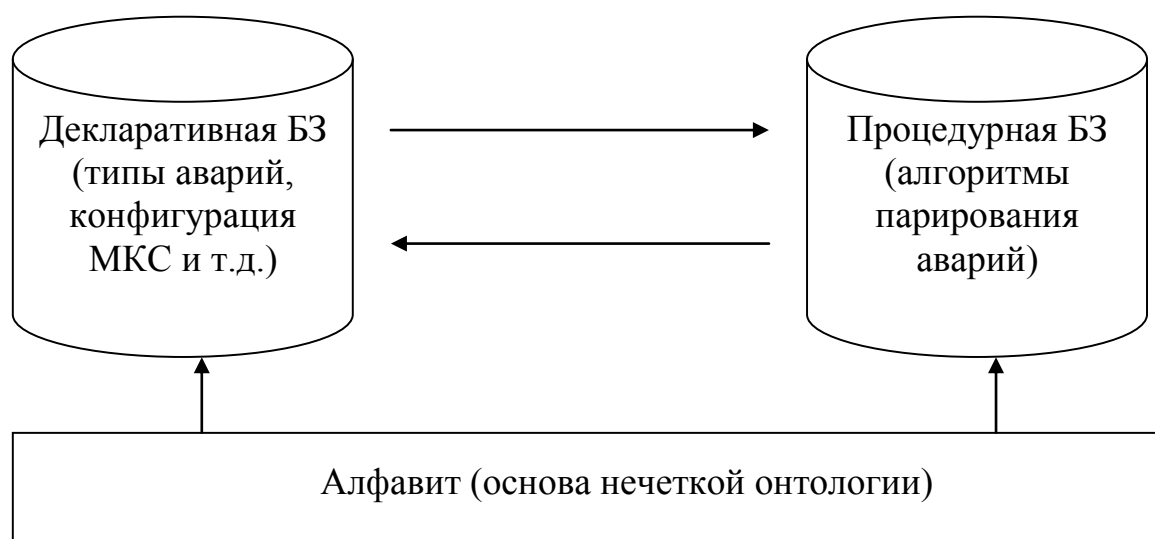


Рис. 1. Упрощенная структура нечеткой онтологии

Система поддержки принятия решений, построенная на основе нечеткой онтологии, будет способна эффективно решать задачи выбора

наилучшего решения из множества альтернатив. Эффективность нечеткой онтологии обеспечивается за счет удовлетворения критериям для проектирования онтологии, а также использования нечеткости, которая позволяет описывать предметную область на естественном языке и упрощать процесс взаимодействия системы с пользователем.

Литература

1. Система поддержки принятия решений // Википедия. URL: [ru.wikipedia.org/wiki/ Система_поддержки_принятия_решений](http://ru.wikipedia.org/wiki/Система_поддержки_принятия_решений) (дата обращения: 23.06.2012).
2. Gruber T.R. A Translation Approach to Portable Ontology Specifications // Knowledge Sharing / Ed. By N. Mars. Amsterdam: IOS Press, 1995.
3. Смирнов А.В., Пашкин М.П., Шилов Н.Г., Леванова Т.В. Онтологии в системах искусственного интеллекта: способы построения и организации (часть 1) // Новости искусственного интеллекта. 2002. № 1 (49).
4. Lopez M. F. Overview of Methodologies for Building Ontologies. Proceedings of the IJCAI-99 Workshop on Ontologies and Problem-Solving Methods (KRR5) / Eds. By V.R. Benjamins, B. Chandrasekaran, A. Gomez-Perez, N. Guarino, M. Uschold. Stockholm, Sweden, 1999.
5. Kayed A., Colomb R.M. Re-engineering Approach to Build Domain Ontologies. Proceedings of the First Asia-Pacific Conference on Web Intelligence (WI-2001). Springer, Lecture Notes in Computer Science, 2001. № 2198.

УДК 510.3

О МЕТОДАХ ТЕОРИИ НЕЧЕТКИХ МНОЖЕСТВ В ИНФОРМАЦИОННЫХ СИСТЕМАХ

Погорелов А.С., Панфилов А.Н.

Южно-Российский государственный технический университет (НПИ), г. Новочеркасск

При разработке нечетких информационных систем часто возникает задача агрегирования входной нечеткой информации. Нечеткость означает, что информация обладает некоторой степенью неопределенности и очень часто является неполной. Из задачи агрегирования нечеткой информации возникает задача восстановления неполных данных. Восстановление необходимо для корректной реализации операции агрегирования. В данной статье проводится обзор статистических методов восстановления неполных данных, а также ставится задача построения новых методов обработки неполной информации на основе математического аппарата теории нечетких множеств и интервальной арифметики.

В настоящее время теория нечетких множеств достаточно интенсивно применяется при разработке информационных систем в различных областях человеческой деятельности. В технических областях можно назвать автоматическое управление высокого уровня, автоматический перевод, системы поддержания целостности баз данных и системы обеспече-

ния безопасности, распознавание изображений и речи, автоматическое проектирование, поиск информации и т. д. В медицине – это диагностика, роботы для ухода за больными, системы здравоохранения. В сфере бизнеса – помощь в принятии экономических решений, маркетинг, различного рода управление и планирование, автоматизация учреждений [1].

Нечеткие системы, независимо от области применения, характеризуются тем, что на вход системы подается нечеткая информация. При этом нечеткость может проявляться по-разному. Например, данные могут обладать некоторой степенью неопределенности, вызываемой такими факторами, как: ошибки в измерениях, при округлении, сбои в работе датчиков и т.д. [2]. Это означает, что данные могут быть частично потеряны, искажены или могут отсутствовать по каким-либо другим причинам. Такая информация называется неполной. При этом часто возникает проблема, когда необходимо оценить ситуацию «в общем» на основе исходных параметров. Эта оценка основывается на операции агрегирования информации, в результате чего на выходе появляется некий обобщенный результат, который получается из совокупности входных параметров.

Из задачи агрегирования нечеткой информации возникает задача восстановления неполных данных. Восстановление необходимо для корректной реализации операции агрегирования. Далее предполагается, что исходные данные представлены в виде таблицы, каждая строка которой представляет собой массив параметров определенного объекта наблюдения. Наиболее распространенными методами обработки неполной информации являются следующие [3]:

1. Изъятие некомплектных строк из таблицы. Строка или столбец таблицы данных называется некомплектным, если в нем отсутствует хотя бы одно значение. Метод применяется при большой размерности таблицы и незначительном количестве пропусков. В других случаях метод ведет к смещенности оценок, поскольку строки с пропущенными значениями содержат новую информацию, необходимую для анализа.

2. Заполнение пропусков средними значениями. В результате применения такого метода несколько значений одного фактора оказываются одинаковыми, что указывает на его низкую точность.

3. Метод ближайших соседей [4]. Находят строки таблицы, которые по определенному критерию являются ближайшими к строке с пропуском. Для его заполнения значения фактора у соседей усредняются с весовыми

коэффициентами, обратно пропорциональными их декартовому расстоянию к строке, которая содержит пропуск.

4. Регрессионный метод [4]. По комплектным данным строится уравнение линейной множественной регрессии и вычисляются пропущенные значения факторов.

5. Метод максимальной правдоподобности и ЕМ-алгоритм [2]. Требует проверки гипотез о распределении значений факторов. Применение осложняется в случае большого количества пропущенных значений фактора.

6. Алгоритм ZET [5]. Главная идея алгоритма заключается в подборе «компетентной матрицы», используя данные из которой дальше находят параметры зависимости, которая используется для прогнозирования пропущенного значения.

7. Алгоритм ZetBraid [5]. Основное отличие от алгоритма ZET заключается в определении оптимального размера «компетентной матрицы».

8. Метод Бартлетта [4]. Неитеративный метод, который применяется для заполнения пропусков в векторе значений результирующей характеристики в допущении, что значения входных факторов являются комплектными.

Все вышеперечисленные методы восстановления пропусков в данных объединяются понятием «статистические методы» [4]. Судя по растущему количеству публикаций, посвященных обсуждаемой проблеме, все большее число исследователей склоняется к тому, что в реальном математическом моделировании наиболее целесообразным подходом можно считать представление исходных данных в виде нечетких множеств или интервальных значений [6]. Отмечается отсутствие достаточно обоснованных математических методов получения и представления исходных данных в виде нечетких множеств. В связи с этим, возникает задача построения новых методов обработки неполной информации на основе математического аппарата теории нечетких множеств и интервальной арифметики. Получение таких эффективных методов позволит решить задачу агрегирования нечеткой информации, которая является достаточно актуальной при проектировании нечетких систем.

Литература

1. Прикладные нечеткие системы: Пер. с япон./ К. Асаи, Д. Ватада, С.Иваи и др.; под редакцией Т. Тэрано, К. Асаи, М. Сугэно. – М.: Мир, 1993. – 368 с., ил.
2. Zoubin Ghahramani and Michael I. Jordan Supervised learning from incomplete data via an EM approach, Department of Brain & Cognitive Sciences Massachusetts Institute of Technology, Cambridge, MA 02139

3. Снитюк В. Е. Эволюционный метод восстановления пропусков данных. Интеллектуальный анализ информации. Межд. конф. – Киев, 2006.
4. Злоба Е., Яцков И. Статистические методы восстановления пропущенных данных // Computer Modelling & New Technologies. – 2002. – Vol. 6. – №1.
5. Загоруйко Н. Г. Методы распознавания и их применение. – М.: Сов. Радио, 1972. – 216 с.
6. Перепелица В.А., Тебуева Ф.Б. Дискретная оптимизация и моделирование в условиях неопределенности данных // «Академия Естествознания». – 2007

УДК 004

РЕАЛИЗАЦИЯ АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ ЭЛЕКТРОННОГО ДОКУМЕНТООБОРОТА ДЛЯ ОТДЕЛА ВНЕШНИХ СВЯЗЕЙ ООО «КЗ «РОСТСЕЛЬМАШ»

Сейтумерова Л.Б., Воробьев С.П.

Южно-Российский государственный технический университет (НПИ), г. Новочеркасск

Приводится описание системных решений, принятых в процессе разработки Автоматизированной системы электронного документооборота. Представлены разработанные диаграммы IDEF0, функциональная структура системы документооборота, структура взаимодействия программного и информационного обеспечения.

В соответствие с общей политикой автоматизации компании ООО «КЗ «Ростсельмаш» перед руководством возникла задача автоматизации работы отдела внешних экономических связей. Так как единственный аналог, полностью удовлетворяющий потребностям отдела, существует на базе SAP ERP 6.0 и соответственно очень дорогой, то было получено задание, разработать автоматизированную систему электронного документооборота. Система предназначена для создания, ведения и обработки рабочих документов, а также для мониторинга процессов, сопряженных со спецификой работы отдела внешних экономических связей. Для более детального представления бизнес-процессов были разработаны диаграммы IDEF0 процесса импорта деталей (рис. 1).

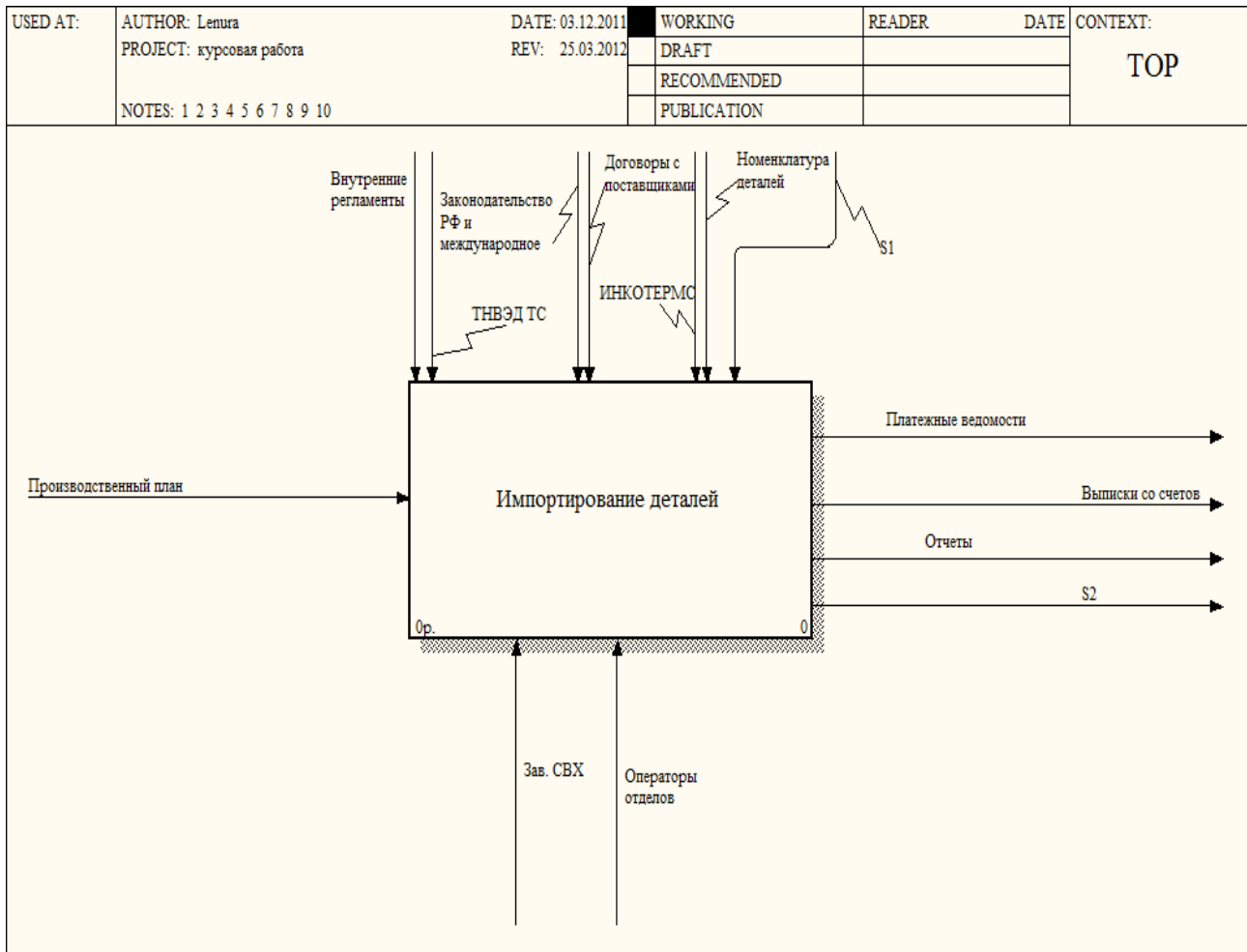


Рис. 1. Контекстная диаграмма «Импортирование деталей»

Контекстная диаграмма в первом приближении декомпозируется на 4 бизнес-процесса (рис. 2):

- Формирование плана импортных поставок
- Планирование отгрузок
- Таможенное оформление
- Выгрузка деталей.

В качестве архитектуры построения АС была выбрана архитектура клиент-сервер: бизнес-логика разделена между клиентской машиной и серверной.

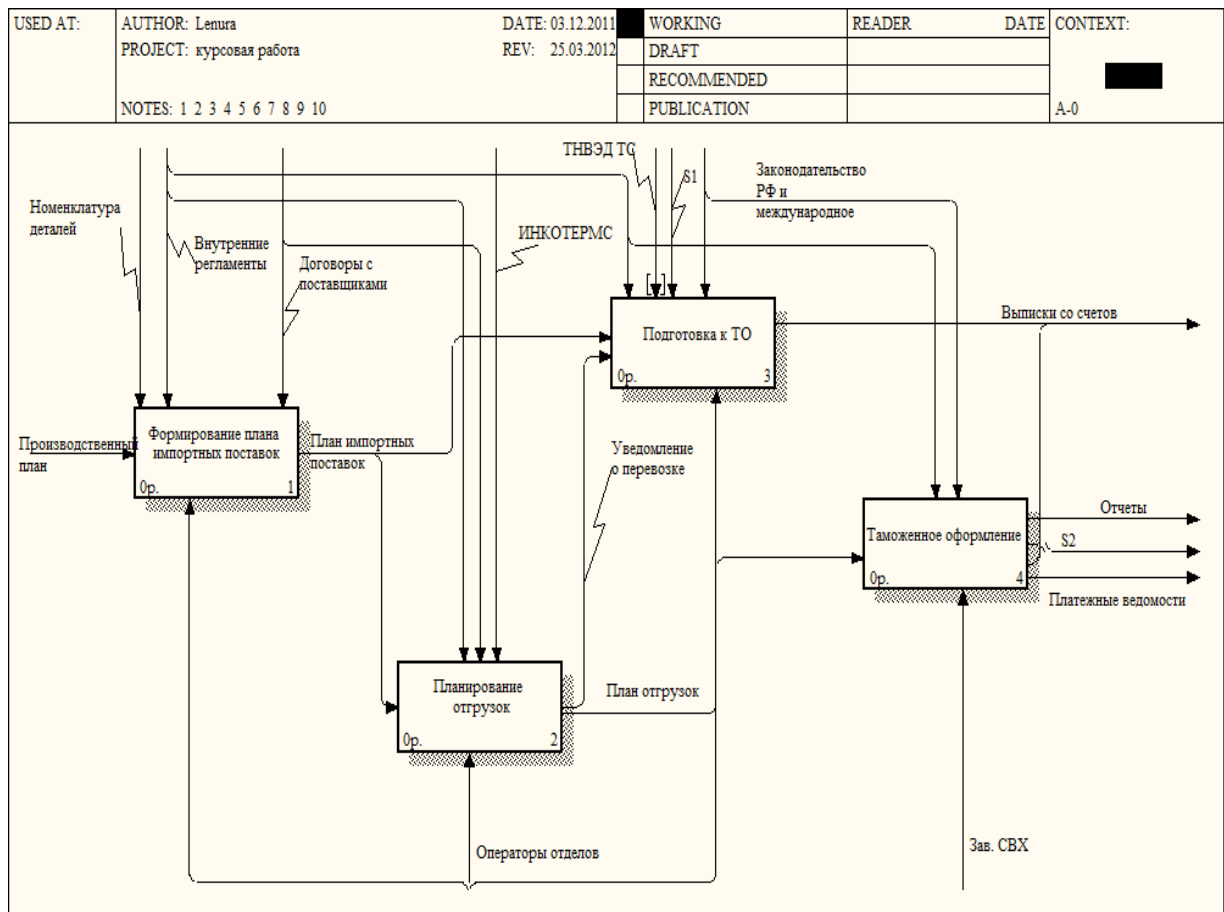


Рис. 2. Декомпозиция контекстной диаграммы

Основными аспектами, опираясь на которые был сделан выбор, явились:

- хранение и обработка производится централизованно;
- снижена нагрузка на сеть, так по сети передаются только SQL-запросы и результаты исполнения транзакции;
- высокая масштабируемость;
- относительно низкие требования к производительности и техническим характеристикам терминалов, и как следствие снижение их стоимости;
- наличие метода доступа ADO.NET, разработанного для архитектуры клиент-сервер.

Данная архитектура обладает существенным недостатком: при выходе из строя сервера БД, система не может функционировать без поддержки информационной части. Однако эта проблема решена с помощью дублирующего сервера, который включится в работу при выходе из строя основного сервера.

Система поддерживает следующий функционал (рис. 3):

- Создание документа или заявки;
- Анализ документа или заказа;
- Исполнение заявки;
- Закрытие;
- Архивирование;
- Поиск;
- Ведение регистрационной карточки документа;
- Хранение версий;
- Отслеживание и оповещение о статусах документов;
- Формирование задач и обозначение сроков их исполнения;
- Отслеживание и оповещение о смене статуса груза;
- Заполнение справочников;
- Автоматическая и ручная актуализация справочников;
- Настройка подключения к серверу БД;
- Настройка подключения к файл-серверу;
- Работа с файлами конфигурации;

На основании предложенного функционала была разработана поддерживающая информационная система, а также программные модули, для взаимодействия с пользователями (рис. 4).

На схеме изображено взаимодействие конкретных программных модулей с таблицами базы данных.

Проект был успешно внедрен на предприятии. Однако не весь функционал в части мониторинга был реализован: оповещение пользователей всплывающими сообщениями, а также хранение версий документов. В дальнейшем планируется доработка и модернизация проекта.

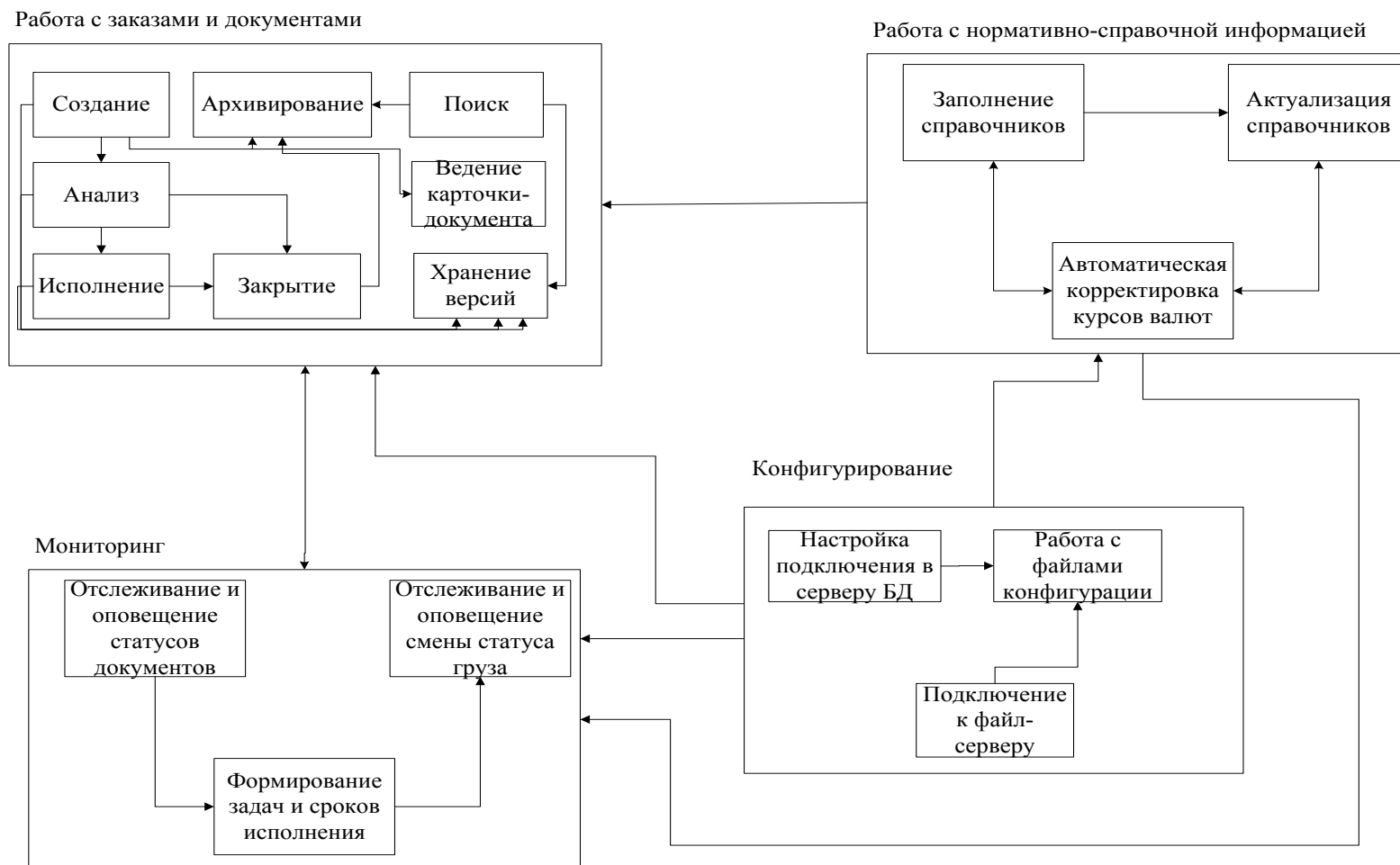


Рис. 3. Функциональная структура системы документооборота

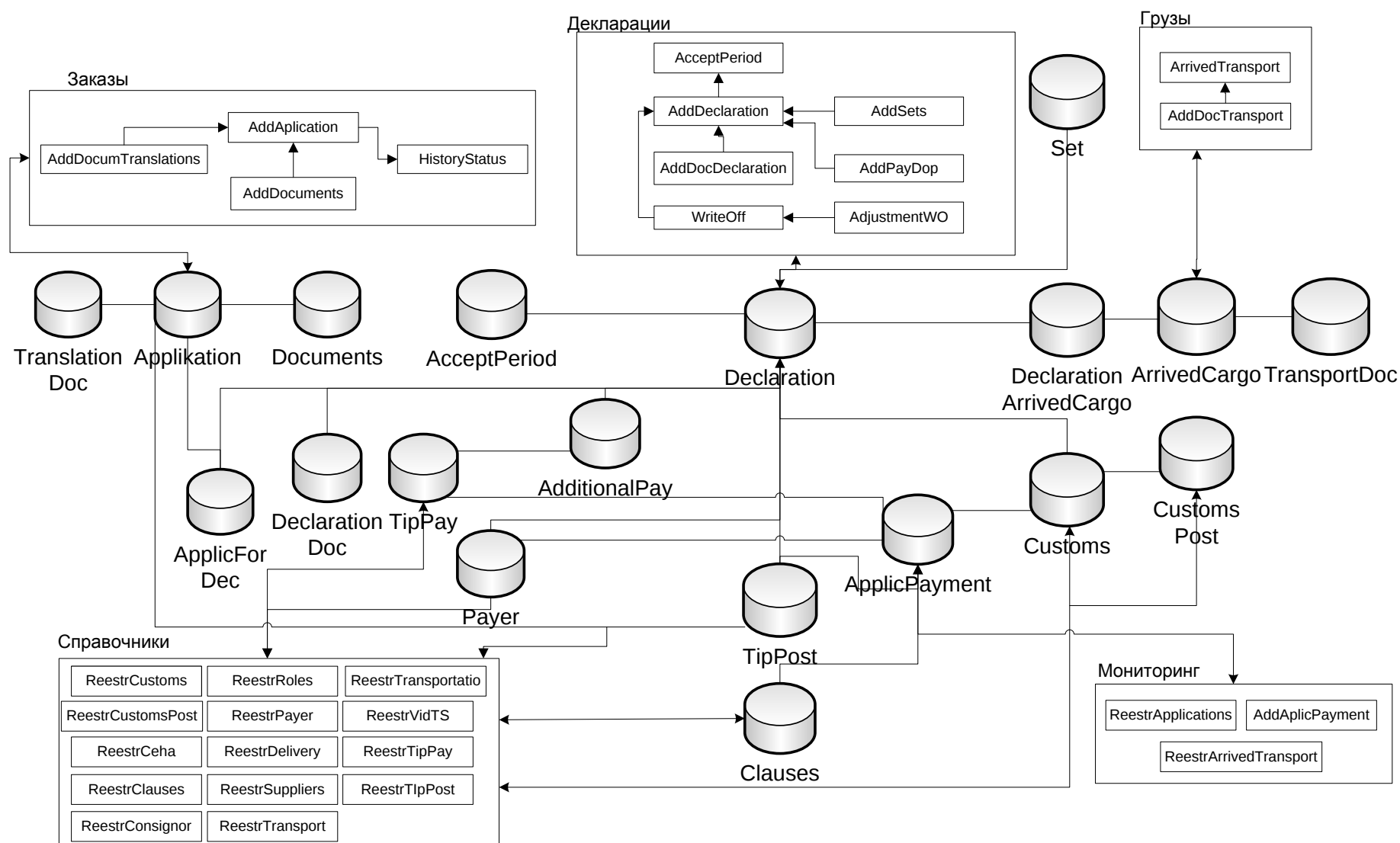


Рис. 4. Структура взаимодействия программного и информационного обеспечения

МАТЕМАТИЧЕСКИЕ МОДЕЛИ СИСТЕМ ЭЛЕКТРОННОГО ДОКУМЕНТООБОРОТА

Сейтумерова Л.Б., Воробьев С.П.

Южно-Российский государственный технический университет (НПИ), г. Новочеркасск

Был проведен обзор математических моделей, используемых при проектировании и реализации систем электронного документооборота.

В связи с интенсивным развитием информационных технологий на первое место выходит задача автоматизации документооборота. Проблемы организации электронного документооборота, внедрения и выбора системы управления документами нашли отражение в трудах В.Э. Баласаняна, С.В. Клименко, С.Л. Кузнецова, М.В. Ларина, Д.А. Романовой и др. Но на настоящий момент времени проблемы проектирования, реализации и внедрения эффективной системы электронного документооборота, которая учитывает специфику бизнес-процессов конкретного предприятия в полном объеме не решена. Существуют отдельные подходы, которые не предоставляют общей универсальной методологии построения и интеграции систем данного класса.

В статье [1] на базе элементов формальной модели композитного документооборота рассматриваются критерии эффективности, которые позволяют с высокой степенью адекватности оценивать проектирование, разработку и внедрения систем электронного документооборота.

В данной работе М.Ю. Круковский предлагает рассматривать СЭДО с точки зрения трех множеств:

- множество участников документооборота $\{Y\}$ – это сотрудники организации, производящие создание, движение, модификацию документов.
- множество действий участников $\{D\}$, представляет собой конечный список возбуждающих воздействий.
- множество состояний документов $\{F\}$, которые могут принимать документы во время жизненного цикла документооборота.

В качестве критерия эффективности для получения корреляций выделенных параметров используются минимаксный критерий. При комбинации сочетаний над элементами тройки множеств $\{Y\}$ $\{D\}$ $\{F\}$ получены критерии эффективности по действиям, по участникам, по состояниям.

В зависимости от приоритетности критерия конкретной СЭДО выделяется один из критериев как основной.

Количественный критерий эффективности документооборота выражается отношением композитной части к совокупности электронной и бумажной составляющих.

$$Eff = \Phi_e \cap \phi_e / (\Phi_e \cup \phi_e)$$

В публикации [2] была проведена разработка комплекса моделей, обеспечивающих информационную поддержку интеграции информационных систем на основе электронного документооборота, как промежуточного программного обеспечения, автоматизирующего управление документами и бизнес-процессами. В частности были предложены следующие основные требования к моделям:

1. Возможность эффективного добавления, редактирования, удаления и работы с данными произвольного (как структурированного, так и слабо структурированного) типа и размера.

2. Возможность совместного использования различными информационными системами (ИС) одной и той же информации, которая должна осуществляться за счёт предоставления системам сервисов для работы с данными.

3. Возможность интеграции ИС за счёт использования стандартизованного описания формата передачи данных и наличия у систем схожих сервисов работы с ним.

4. Возможность объединения внутренних прикладных процессов каждой ИС за счёт создания в СЭД типовых бизнес-процессов.

5. Возможность автоматического определения метаданных из содержимого текстового документа на основе построении нейронной сети Кохонена, применяемой для классификации образов путём нахождения близости параметров образа к ядру каждого класса., а также определения метаданных из описания, представленного в специальном формате (например, в XML формате).

Ядро класса для метаданного имеет вид:

$$C = (k, r, \{s_1, \dots, s_v\} \{f_1, \dots, f_u\} \{t_1, \dots, t_w\})$$

где k – определяющий признак класса, r – позиция признака в тексте, $\{s_1, \dots, s_v\}$ – ключевые слова, v – количество ключевых слов, $\{f_1, \dots, f_u\}$ – ключевые фразы, u – количество ключевых фраз, $\{t_1, \dots, t_w\}$ – шаблоны, соответствующий синтаксису регулярных выражений, w – количество шаблонов.

В статье [3] рассмотрены способы построения аналитических моделей программных систем защиты информации в автоматизированных системах электронного документооборота. Также приведены аналитические выражения для оценки «защищенности» систем.

В качестве основы для конструирования показателя эффективности ПСЗИ принимают соотношения временных характеристик системы и временных характеристик процесса ее вскрытия. Под временем $\phi(di)$ обеспечения защитных функций ПСЗИ понимают время с момента обращения к ПСЗИ до окончания реализации ею своих функций по данному обращению. Защитные функции ПСЗИ считаются реализованными своевременно, если время $\phi(di)$ не превышает некоторой максимально допустимой величины $\phi(m)$, обусловленной стратегией вскрытия ПСЗИ злоумышленником, т.е. при выполнении неравенства:

$$\phi(di) \leq \phi(m)$$

В каждом отдельном случае $\phi(m)$ имеет конкретное максимальное значение. Обе входящие в неравенство величины являются случайными, поэтому выполнение или не выполнение этого неравенства есть случайное событие, вероятность которого $P \phi(di) \leq \phi(m)$ представляет собой среднее количество ситуаций, когда ПСЗИ своевременно реализуют свои функции в течении интервала ΔT времени эксплуатации автоматизированной системы документооборота относительно общего числа таких ситуаций, то есть

$$P(\phi_{di} \leq \phi_m) = \frac{1}{M} > \sum_{k=1}^k \delta_k, \quad \text{где}$$

$$\delta_k = \begin{cases} 1, & \text{при } \tau_{(di)k} \leq \tau_{(m)k} \\ 0, & \text{при } \tau_{(di)k} > \tau_{(m)k} \end{cases}$$

где $\tau_{(di)k}$ – время реализации функций ПСЗИ при k -ой, $k = 1$, попытке ее вскрытия; $\tau_{(m)k}$ – максимально допустимое время реализации соответствующих шагов стратегии вскрытия ПСЗИ при k -ой попытке.

В статье [4] предлагают нестандартный подход к организации документооборота предприятия на основе выделения ядра системы - хранилища электронных документов, представленное в виде тематического дерева документов и таблицы адресных указателей, связывающих листья дерева с физическими файлами. В общем случае дерево документов описывается с позиций теории множеств

$$G = \{V, D, R\},$$

где G – граф дерева документов, $V_i \in \{V\}$ – множество вершин дерева, $D_i \in \{D\}$ – множество листьев дерева (связей с документами), $R_i \in \{R\}$ – множество связей между вершинами дерева.

Для выполнения операций над деревом документов используется аппарат теории графов, для чего дерево сводят к иерархическому графу.

Один и тот же документ в таком графе может быть связан с несколькими вершинами $Di \Rightarrow \{V\} \neq$, при этом в одну и ту же вершину может входить несколько листьев (документов) $\{D\} \neq \Rightarrow Vi$. В этом случае возникает сложная структура графа с отношениями между вершинами и листьями "многие - ко - многим". Прямая организация такой структуры в традиционных базах данных невозможна. В системе эта структура реализована непосредственно через таблицу связей листьев с вершинами графа.

Помимо представления дерева документов в виде графа используется матричное представление, которое позволяет выполнять некоторые действия над деревом с использованием теории матричных вычислений.

Рассмотренные элементы математических моделей документооборота позволяют создать информационные системы управления документами в различных предметных областях, но требуют доработки с целью формирования единого подхода к проектированию электронного документооборота.

Литература

1. Круковской М.Ю. Критерий эффективности систем электронного документооборота // Системи підтримки прийняття рішень. Теорія і практика. – 2005. – С. 107-111
2. Завозкин С.Ю. Информационное обеспечение интеграции информационных систем на основе СЭДО: автореферат дисс. на соискание ученой степени кандидата техн. Наук. – Кемерово: Кемеровский государственный университет, 2007. – 18 с.
3. Даурцев А.В. Разработка математических моделей оценки показателей эффективности программных систем защиты информации (ПСЗИ) в автоматизированных системах электронного документооборота // Вестник Воронежского государственного технического университета. – 2011. – № 3. – С. 92-96
4. Середа С.Н., Середа Ю.А. Модели автоматизации документооборота // Методы и устройства передачи и обработки информации. – 2002. – № 2. – С. 131-134.

УДК 004

СТРУКТУРНЫЕ ПРЕОБРАЗОВАНИЯ В РАСЧЕТАХ НАДЕЖНОСТИ СИСТЕМ

Сергеев Д.А.

Южно-Российский государственный технический университет (НПИ), г. Новочеркасск

Изложен вариант количественной оценки проектного варианта на этапе структурного синтеза системы.

В расчетах надежности коммуникационных сетей важным этапом является анализ структур, определенных на множестве их свойств и пред-

ставляемых графовыми моделями [1]. В общем случае такие модели описываются С-структурами (наименее определенными), поэтому возникает необходимость декомпозиции модели на базовые в смысле описания функционирования системы элементы и преобразование их в Р-структуры (наиболее уточненное описание). В качестве таковых используют клики графа и соответствующие им Р-структуры. Построив графовую модель, обосновывают процедуру преобразования её в расчетную модель надежности. Одной из широко применяемых процедур является построение структурных схем надежности [2]. Примером служит известное преобразование «треугольник – звезда», изображенное на рис. 1.

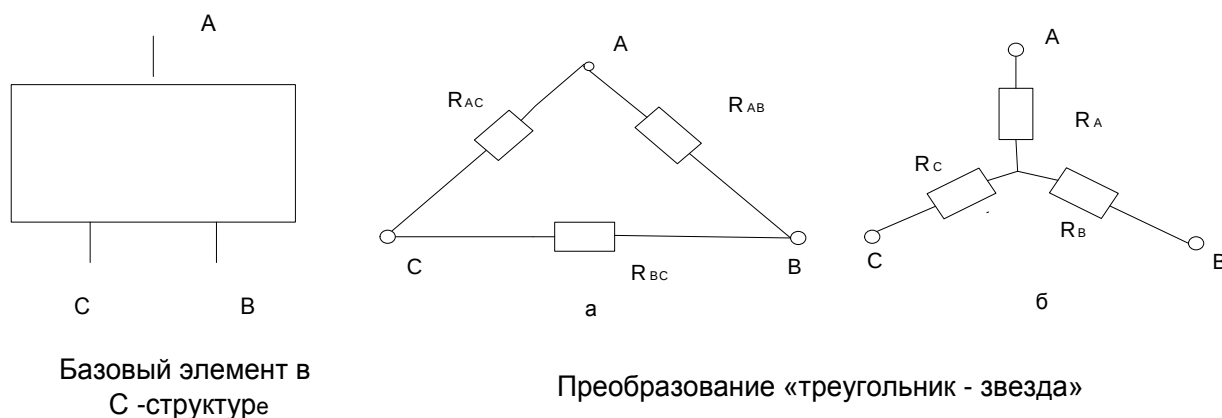


Рис. 1. Пример структурного преобразования

Базовым элементом является «треугольник», соответствующая ему Р-структура – «звезда».

Так как работоспособность участка ССН между узлами не должна зависеть от способа представления, то можно записать систему соотношений между показателями надежности, используемых в них:

$$\begin{aligned} P_A P_B &= P_{AB} + P_{BC} P_{AB} - P_{AB} P_{BC} P_{AC}; \\ P_C P_B &= P_{BC} + P_{AC} P_{AB} - P_{AB} P_{BC} P_{AC}; \\ P_C P_A &= P_{AC} + P_{BC} P_{AB} - P_{AB} P_{BC} P_{AC}. \end{aligned} \quad (1)$$

Для получения формул пересчета показателей надежности при решении системы уравнений (1) вводятся допущения, ограничивающие применение полученных формул. Так в [3] получена формула $P_A = \sqrt{P_{AB} + P_{AB}^2 - P_{AB}^3}$, которая справедлива при условии: $P_{AB} = P_{BC} = P_{AC}$.

В общем случае эта формула не применима.

Решение системы (1) должно соответствовать условиям:

$$0 \leq P_{AB} \leq 1; \quad 0 \leq P_{BC} \leq 1; \quad 0 \leq P_{AC} \leq 1; \quad 0 \leq P_A \leq 1; \quad 0 \leq P_B \leq 1; \quad 0 \leq P_C \leq 1 \quad (2)$$

Вводя обозначения:

$$\begin{aligned} A_1 &= P_{AB}P_{BC} P_{AB} - P_{AB} P_{BC} P_{AC}; \\ A_2 &= P_{BC} + P_{AC} P_{AB} - P_{AB} P_{BC} P_{AC}; \\ A_3 &= P_{AC} + P_{BC} P_{AB} - P_{AB} P_{BC} P_{AC}, \end{aligned}$$

выпишем формулы преобразования «треугольник» $\rightarrow$ «звезда»

$$P_A = \left(\frac{A_1 A_3}{A_2} \right)^{\frac{1}{2}}; \quad P_B = \left(\frac{A_1 A_2}{A_3} \right)^{\frac{1}{2}}; \quad P_C = \left(\frac{A_2 A_3}{A_1} \right)^{\frac{1}{2}} \quad (3)$$

С целью упрощения расчетов выполним приведение формул (3) к одному показателю P_{AB} :

$$\begin{aligned} \frac{P_{AB}}{P_{BC}} = a; \quad \frac{P_{AB}}{P_{AC}} = b; \quad \frac{P_{AB}}{P_{AC}} = ab; \quad K_I &= \frac{\left(1 + bP_{AB} - \frac{P_{AB}^2}{a}\right) \left(1 + \frac{P_{AB}}{a^2 b} - \frac{P_{AB}^2}{a^2 b}\right)}{\left(1 + P_{AB}/b - \frac{P_{AB}^2}{ab}\right) b}; \\ K_{II} &= b \frac{\left(1 + \frac{P_{AB}}{b} - \frac{P_{AB}^2}{ab}\right) \left(1 + \frac{P_{AB}}{a^2 b} - \frac{P_{AB}^2}{a^2 b}\right)}{\left(1 + bP_{AB} - \frac{P_{AB}^2}{a}\right)}; \quad K_{III} = \frac{\left(1 + bP_{AB} - \frac{P_{AB}^2}{a}\right) \left(1 + \frac{P_{AB}}{b} - \frac{P_{AB}^2}{ab}\right)}{\left(1 + \frac{P_{AB}}{a^2 b} - \frac{P_{AB}^2}{a^2 b}\right) a^2 b}. \end{aligned}$$

С учетом указанных преобразований формулы (3) принимают вид:

$$P_A = (P_{AB} K_I)^{\frac{1}{2}}; \quad P_B = (P_{AB} K_{II})^{\frac{1}{2}}; \quad P_C = (P_{AB} K_{III})^{\frac{1}{2}}$$

Анализ приведенных формул показывает, что для выполнения условий (2) на параметры a и b накладываются следующие ограничения при

$$\begin{aligned} 0 \leq P_{AB} \leq 1, \quad ab \geq 1, \quad \frac{1}{a} \leq b \leq 1 \\ \text{при } ab \leq 1, \quad a \leq 1, \quad b \leq \frac{1}{a}, \quad 0 \leq P_{AB} \leq ab. \end{aligned}$$

Для обратного перехода используются формулы:

$$P_{AB} = \frac{P_A^2}{K_I}; \quad P_{BC} = \frac{P_B^2}{a K_{II}}; \quad P_{AC} = \frac{P_C^2}{ab K_{III}}$$

В таблице 1 приведены значения показателей надежности для перехода «треугольник» $\rightarrow$ «звезда» при $a = 1$, $ab = 1$

Таблица 1

P_{AB}	0,9	0,8	0,7	0,6	0,5	0,4	0,3
----------	-----	-----	-----	-----	-----	-----	-----

P_A	0,990	0,963	0,920	0,863	0,791	0,704	0,603
-------	-------	-------	-------	-------	-------	-------	-------

В таблице 2 приведены значения показателей для различных значений a и b .

Таблица 2

P_{AB}	$P_A;$ $P_B; P_C$	a	b	P_A	P_{AB}	P_B	P_{BC}	P_C	P_{AC}
0,85	0,95; 0,99; 0,96	1,06 1,06	1 1	0,955	0,85	0,991	0,8	0,956	0,8
0,7	0,874; 0,924; 0,834	1,55 1,16	0,75 1	0,734	0,7	0,903	0,6	0,77	0,6
0,8	0,646; 0,709; 0,875	1,33 1,33	0,5	0,787	0,6	0,982	0,45	0,924	0,9
0,7	0,681; 0,636; 0,595	1,4 1,66	0,55 0,43	0,641	0,5	0,818	0,3	0,970	0,7
0,6	0,651; 0,855; 0,826	1,33 1,75	0,5 0,5	0,646	0,7	0,709	0,4	0,875	0,8
0,7	0,787; 0,982; 0,924	1,4 1,33	0,63 0,86	0,678	0,8	0,850	0,6	0,923	0,7

Другой вариант базового элемента приведен на рис. 2.

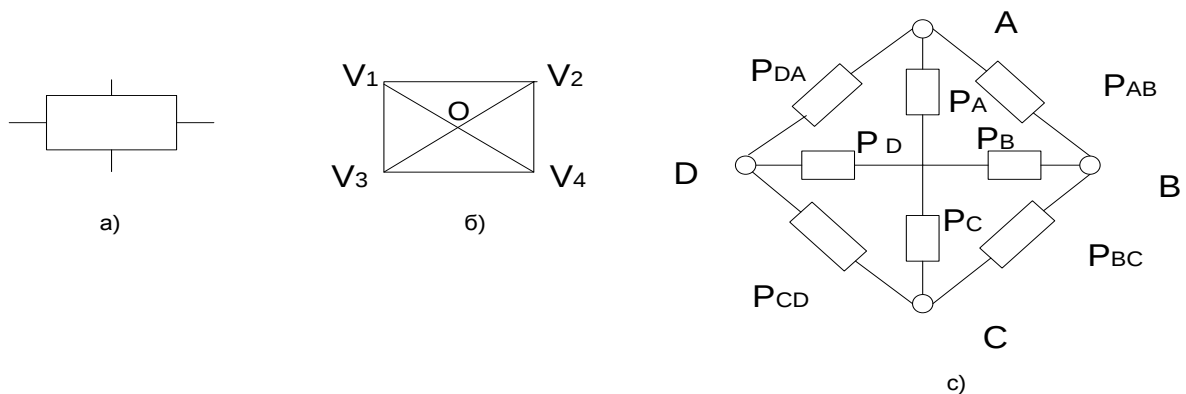


Рис. 2. Преобразование «четырёхугольник – звезда»

На рис. 2, а приведена С – структура базового элемента, 2, б – соответствующая ей Р – структура; 2, с – схема преобразования.

Аналогично (1) выписываем систему уравнений баланса показателей надежности относительно узлов А, В, С, D:

$$\begin{aligned} P_A P_B &= P_{AB} + P_{BC} P_{CD} P_{DA} - P_{AB} P_{BC} P_{CD} P_{DA} = A_1; \\ P_B P_C &= P_{BC} + P_{AB} P_{CD} P_{DA} - P_{AB} P_{BC} P_{CD} P_{DA} = A_2; \\ P_C P_D &= P_{CD} + P_{BC} P_{AB} P_{DA} - P_{AB} P_{BC} P_{CD} P_{DA} = A_3; \\ P_D P_A &= P_{DA} + P_{AB} P_{CD} P_{BC} - P_{AB} P_{BC} P_{CD} P_{DA} = A_4. \end{aligned} \quad (4)$$

Решение должно соответствовать условию $P_A \leq 1$; $P_B \leq 1$; $P_C \leq 1$; $P_D \leq 1$.

Выписывая $\frac{P_A}{P_B} + \frac{P_A}{P_D} = A_1 + A_4$, получим $P_A \frac{P_B + P_D}{P_B P_D} = A_1 + A_4$ (4).

Учитывая $\frac{P_B}{P_D} = \frac{A_2}{A_3}$; $P_A P_D = A_4$, после преобразования (4), получим

$$P_D = \sqrt{\frac{\left(\frac{A_3}{1+\frac{A_3}{A_2}}\right)}{\left(\frac{A_1}{1+\frac{A_1}{A_4}}\right)}}; \quad P_A = A_4 / \sqrt{\frac{\left(\frac{A_3}{1+\frac{A_3}{A_2}}\right)}{\left(\frac{A_1}{1+\frac{A_1}{A_4}}\right)}}; \quad P_B = \frac{A_2}{A_3} \sqrt{\frac{\left(\frac{A_3}{1+\frac{A_3}{A_2}}\right)}{\left(\frac{A_1}{1+\frac{A_1}{A_4}}\right)}}; \quad P_C = A_3 / \sqrt{\frac{\left(\frac{A_3}{1+\frac{A_3}{A_2}}\right)}{\left(\frac{A_1}{1+\frac{A_1}{A_4}}\right)}}.$$

Для получения правил обратного перехода выразим вероятности P_{IJ} через отношения A_J/A_I .

Формируя отношение разностей первого и четвертого, первого и третьего уравнений в (4), выпишем:

$$\frac{A_1 - A_4}{A_1 - A_3} = \frac{P_{AB} - P_{DA}}{P_{AB} - P_{CD}}.$$

Отсюда получаем $\frac{A_4}{A_1} = \frac{P_{DA}}{P_{AB}}$; $\frac{A_3}{A_1} = \frac{P_{CD}}{P_{AB}}$. Учитывая, что

$$A_1 - A_2 = \frac{P_{AB} - P_{BC}}{1-b}; \quad A_4 - A_3 = \frac{P_{DA} - P_{CD}}{1-a}; \quad b = P_{CD} P_{DA}; \quad a = P_{AB} P_{BC} \text{ получаем:}$$

$$P_{AB} = \frac{A_1}{A_4} P_{DA}; \quad P_{CD} = \frac{A_2}{A_4} P_{DA}; \quad P_{DA} = \frac{\left(\frac{1-\frac{A_3}{A_4}}{1-\frac{A_2}{A_4}}\right)(1-a)}{A_4 \left(\frac{1-\frac{A_2}{A_4}}{1-\frac{A_3}{A_4}}\right)};$$

$$P_{BC} = A_1 (1-a) \frac{\left(\frac{1-\frac{A_3}{A_4}}{1-\frac{A_2}{A_4}}\right)}{\left(\frac{1-\frac{A_2}{A_4}}{1-\frac{A_3}{A_4}}\right)} - (1-b)(A_1 - A_2)$$

Как инструмент исследования структурных свойств системы рядом преимуществ обладает методология IDEF0 [4]. Она позволяет формировать иерархическую структуру объяснительных моделей-диаграмм. В таких моделях базовым элементом описания функционирования системы является блок, характеризуемый сущностями «вход», «выход», «управление», «исполнитель». Однако в стандарте отсутствуют средства количественной оценки качества проекта. Для осуществления такой оценки следует определить значения показателей надежности указанных сущностей на нижнем уровне («действий») и, применяя приведенные преобразования, рассчитать показатели надежности верхнего уровня («деятельности»).

Литература

1. Дж. Клир. Системология. Автоматизация решения системных задач. – М.: «Радио и связь», 1990
2. Гост Р 51901.14-2005 Метод структурной схемы надежности.
3. Голинкевич Т.А. Прикладная теория надежности. – М.: ВШ, 1985
4. Р 50.1.028-2001. Методология функционального моделирования.

УДК 004:629.735.45

ИНФОРМАЦИОННЫЕ СИСТЕМЫ В ОБЛАСТИ ИСПЫТАНИЙ ВЕРТОЛЕТНЫХ ДВИГАТЕЛЕЙ

Черноморова Т.С., Богучарская А.Н.

Южно-Российский государственный технический университет (НПИ), г. Новочеркасск

Рассматривается оригинальная структура информационной системы для автоматизации испытаний авиационных двигателей, целью внедрения которой является повышение точности и достоверности оценки параметров двигателей, а также уменьшение времени требуемого для их испытаний.

Испытание вертолетных двигателей является неотъемлемой частью как при их производстве, так и после определенного времени пребывания в эксплуатации. Любое испытание проводится на испытательной станции с использованием испытательного стенда. Под испытательным стендом авиационного газотурбинного двигателя (ГТД) понимается единый комплекс средств и испытательного оборудования, предназначенный для проведения испытаний авиационных ГТД в наземных или имитируемых эксплуатационных условиях.[1]

В современном мире неприемлемо вручную снимать многочисленные показания с датчиков, вести протоколы испытаний, производить рас-

четы и формировать графики. Необходимо создать автоматизированное рабочее место оператора-испытателя. Создание информационной системы (ИС) позволит сократить время сбора и обработки данных, а также снизить количество ошибок, связанных со снятием параметров с датчиков и расчетами, а также повысить производительность труда.

Так как информация об испытаниях вертолетных двигателей является закрытой информацией, при проектировании ИС следует учитывать следующее:

- необходимо организовать разграничение прав пользователей (роль администратора и роль оператора-испытателя);
- необходимо организовать шифрование данных;
- не использовать сеть Интернет для передачи данных.

Так как на одном предприятии одновременно проводится сразу несколько испытаний, необходимо использовать для организации системы архитектуру клиент-сервер. Для ИС в данном случае больше подходит архитектура с толстым клиентом, так он обладает широким функционалом, обеспечивает режим многопользовательской работы, предоставляет возможность работы даже при обрывах связи с сервером (что очень важно при непрерывном снятии параметров с датчиков), имеет возможность подключения к банкам без использования сети Интернет, обладает высоким быстродействием. [2] Возможная схема расположения технических средств на испытательной станции представлена на рис. 1.



Рис. 1. Возможное расположение технических средств на испытательной станции

Таким образом, к техническим средствам программно-аппаратного комплекса для проведения испытаний вертолетных двигателей относятся:

- сервер;
- источник бесперебойного питания;
- рабочая станция;
- среда передачи данных;
- коммутатор;
- сетевой адаптер;
- принтер;
- датчики;
- вторичные приборы;
- преобразователь интерфейсов.

Одной из наиболее надежных СУБД является Microsoft SQL Server 2008, также SQL Server обеспечивает высочайший уровень безопасности и масштабируемости для критически важных приложений.

Для реализации клиентского приложения ИС предлагается интегрированная среда разработки Microsoft Visual Studio 2010, язык разработки C#. В качестве модели доступа к данным предлагается использовать ADO.NET.

Для связи проектируемой ИС с аппаратным комплексом испытательного стенда необходимо использовать OPC-сервер и OPC-клиент. Возможная схема структуры программного обеспечения ИС представлена на рис. 2.

Программа OPC-сервер выполняет непосредственное взаимодействие с аппаратурой, используя аппаратные интерфейсы компьютера. [3] OPC-сервер обеспечивает сбор данных, передачу команд управления, диагностику каналов связи и т.д. OPC-сервер создает программные интерфейсы, обеспечивающие доступ к данным.

Программа OPC-клиент получает данные через интерфейс сервера и выполняет их комплексную обработку – использует для визуализации, строит графики, выводит на печать, сохраняет на диске и т.д. Но так как OPC-клиент хранит данные недолгий срок необходимо связать его с проектируемой ИС.

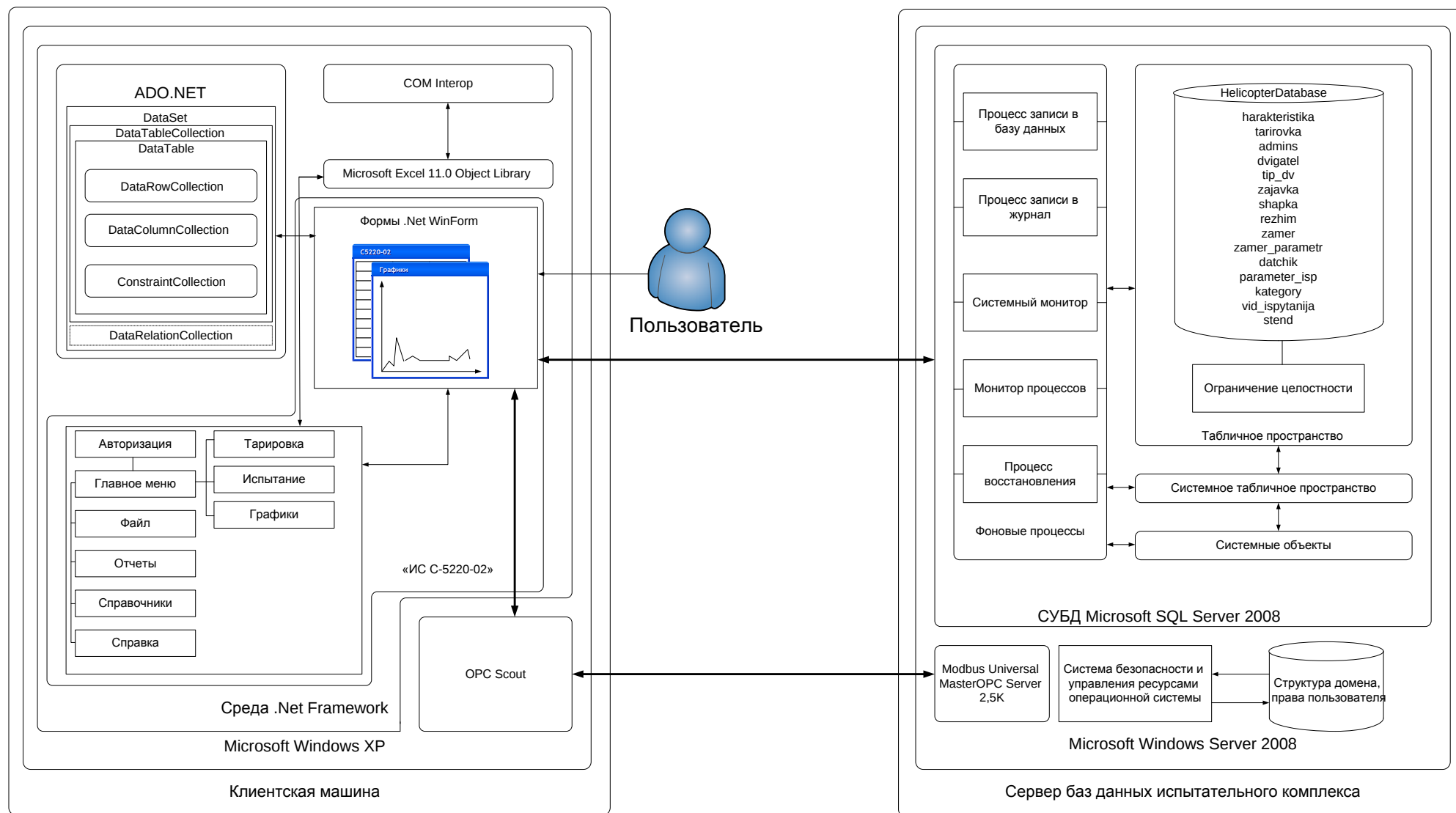


Рис. 2. Структура программного обеспечения информационной системы

Снятие параметров происходит с датчиков с аналоговым интерфейсом. Далее к датчикам подключается вторичный прибор для преобразования аналогового сигнала в цифровой. По токовой петле вторичные приборы подключаются к преобразователю интерфейсов. Далее с преобразователя по RS-232 интерфейсу данные поступают в компьютер на com-порт. Схема подключения вторичных приборов к адаптеру и персональному компьютеру представлена на рис. 3.

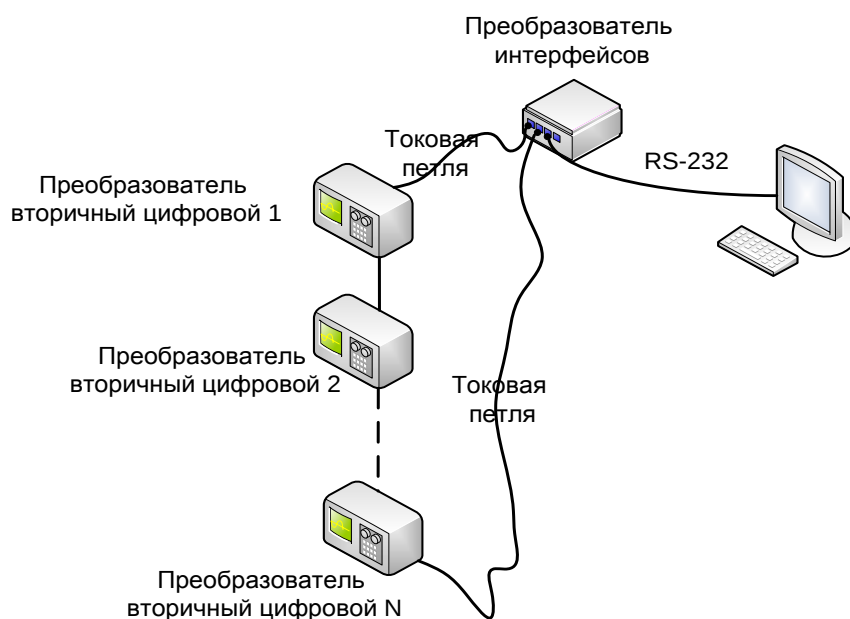


Рис. 3. Схема подключения вторичных приборов к преобразователю интерфейсов и ПК

Для правильного функционирования системы необходимо произвести настройки в OPC-клиенте и OPC-сервере. Требуется задать адрес датчика и тип снимаемого параметра, а также присвоить ячейку, куда будут записываться данные.

Была спроектирована «ИС С-5220-02». Система записывает данные с датчиков в базу, производит обработку данных (строит графики зависимости параметров от времени, формирует протоколы испытаний, заявки на ремонт), сигнализирует об отклонениях параметров от нормальных значений. Были разработаны тесты, которые позволили апробировать работу системы на основании реальных значениях параметров из протоколов испытаний двигателя за прошедшие несколько лет.

Литература

1. <http://www.bestreferat.ru/referat-193624.html>
2. http://ru.wikipedia.org/wiki/Толстый_клиент
3. http://www.novosoft.by/?page=b_opc

УДК 37.01:004.02

СПОСОБ ПОВЫШЕНИЯ ЭФФЕКТИВНОСТИ КОМПЬЮТЕРНОГО ТЕСТИРОВАНИЯ

Чмыхов А.А., Михайлов А.А.

Южно-Российский государственный технический университет (НПИ), г. Новочеркасск

Рассмотрены проблемы эффективности педагогических тестов и предложены рекомендации по повышению эффективности компьютерного тестирования за счет использования адаптивных алгоритмов построения вариантов тестовых заданий.

В настоящее время компьютерное тестирование широко применяется в образовательном процессе при экзаменационной проверке знаний учащихся. Оценка эффективности выполнения задания проводится путем проверки совпадения номеров верного ответа и варианта, выбранного тестируемым. Комплекты заданий, включенных в отдельно взятые варианты теста, как правило определяются заранее. Такой подход не всегда является эффективным, так как тестовые задания могут не соответствовать уровню подготовки испытуемых, что может исказить результаты тестирования.

Целью данной статьи является выработка рекомендаций по формированию тестовых заданий, которые позволят повысить эффективность оценки знаний посредством проведения тестирования.

Проблема эффективности педагогических тестов и тестовых заданий является частью общей проблемы эффективности форм и методов педагогической деятельности. Эффективным можно назвать тест, который лучше, чем другие тесты, измеряет знания студентов интересующего уровня подготовленности, с меньшим числом заданий, качественнее, быстрее, дешевле, и все это – по возможности, одновременно.

При определении эффективности теста обращается внимание на два ключевых элемента – это число заданий теста и уровень подготовленности студентов. Если из какого-либо теста с большим числом заданий сделать оптимальный выбор меньшего числа, то может образоваться система, не уступающая заметно по своим свойствам тесту со сравнительно большим числом заданий. Тест с меньшим числом заданий в таком случае можно называть сравнительно более эффективным [1].

Помимо этого, эффективность теста можно оценить с точки зрения соответствия уровня его трудности уровню подготовленности тестируемых в данный момент студентов. С этой точки зрения эффективный тест - это тест, точно соответствующий по трудности заданий уровню подготовленности испытуемых [2]. Очевидно, что ни один заранее сделанный тест ни-

когда не бывает в точности соответствующим уровню подготовленности конкретных испытуемых. Они все разные, и потому для эффективного измерения требуются тесты разного уровня трудности.

С учетом вышесказанного, для повышения эффективности тестирования предлагаются следующие решения:

1. использовать адаптивные алгоритмы построения теста, которые дадут возможность изменять уровень трудности заданий непосредственно во время тестирования в зависимости от ответов, данных студентом;

2. для определения начального уровня трудности заданий использовать уровень текущей успеваемости студента по дисциплине, по которой проводится тестирование.

Литература

1. Аванесов В.С. Определение педагогического теста // Управление школой. – 1999. – № 28. – С. 33
2. Аванесов В.С. Научные проблемы тестового контроля знаний. – М.: Исследовательский центр, 1994. – С. 112

Чмыхов А.А., аспирант кафедры ИИСТ, e-mail: chmikhov@gmail.com

Михайлов А.А., д.т.н., профессор кафедры ИИСТ, e-mail: mih01@mail.ru

УДК 004.03

ОБ ОДНОМ ПОДХОДЕ К ПОСТРОЕНИЮ РАЗНОРОДНЫХ СИСТЕМ В ЕДИНОМ ИНФОРМАЦИОННОМ ПРОСТРАНСТВЕ

Чмыхов А.А., Панфилов А.Н., Орлов С.Б.

Южно-Российский государственный технический университет (НПИ), г. Новочеркасск

Рассмотрено одно из возможных решений проблемы расширения корпоративных информационных систем на примере разрабатываемой в настоящее время в ЮРГТУ (НПИ) информационной системы «Учебный процесс».

В настоящее время в крупных организациях зачастую применяются СУБД для учета, хранения и анализа больших объемов данных. В большинстве случаев для взаимодействия с информационной базой используется «интеллектуальный» клиент. Это проверенное годами решение для устойчивых, стабильных систем, которые не претерпевают изменений в процессе функционирования. Однако при расширении функциональности информационной системы такой подход становится непригодным в силу большой трудоемкости процесса доработки и перенастройки клиента. Одним из спо-

способов преодоления этой проблемы является создание качественно нового интерфейса к информационной базе, а именно web-интерфейса [1]. Рассмотрим такой подход на примере разрабатываемой в настоящее время в ЮРГТУ(НПИ) информационной системы «Учебный процесс».

Схема взаимодействия компонентов разрабатываемой системы приведена на рис. 1.

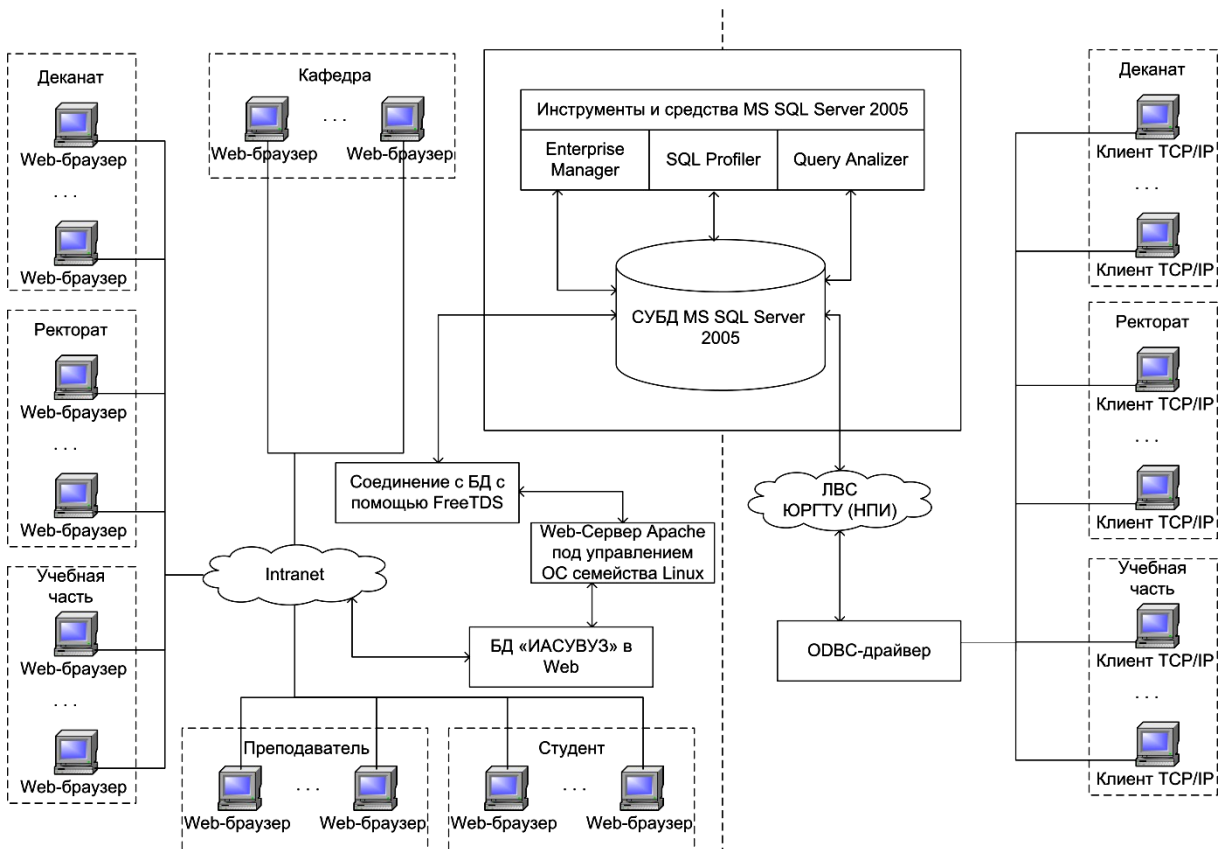


Рис. 1. Схема взаимодействия компонентов системы

Как видно из рисунка, к единой информационной базе доступ осуществляется двумя различными путями: посредством TCP/IP-соединения с помощью ODBC-драйвера для «интеллектуального» клиента и через web-интерфейс для «тонкого» клиента. Это означает, что в едином информационном пространстве будут одновременно функционировать две разнородные информационные системы, дополняющие друг друга.

Рассмотрим представленную схему более подробно.

В основе системы лежит информационная база, представленная реляционной базой данных, управляемой СУБД MS SQL Server 2005. База содержит информацию о студентах, изучаемых дисциплинах, учебных планах, подразделениях университета и многое другое.

В правой части схемы изображен принцип подключения TCP/IP-клиентов посредством ODBC-драйвера. Эта часть системы уже разработана и внедрена. Это типичный пример системы с «толстым» (интеллектуальным) клиентом. Системы такого типа принято использовать в тех случаях, когда количество клиентов невелико, клиентские рабочие станции имеют достаточную производительность для работы интеллектуального клиента, а канал передачи данных между клиентом и сервером имеет достаточную пропускную способность для оперативной передачи больших объемов данных от сервера к клиенту. При этом если стоит задача расширения функциональности системы, то необходимо модернизировать клиентское приложение, а потом устанавливать и настраивать его заново на каждом рабочем месте. В случае расширения системы это может стать весьма трудоемкой задачей. От такой концепции построения системы решено отказаться, либо использовать ее ограниченно. Для расширения функциональности и масштабов системы предложено перейти к концепции системы с «тонким клиентом» и вынесением логики обработки данных на отдельный сервер.

Предполагаемый вариант структуры разрабатываемой информационной системы изображен на левой части схемы (рисунок 1). Это организация доступа к информационной базе посредством web-интерфейса [2]. Остановимся на описании такой структуры.

Для обеспечения доступа к базе данных посредством web-интерфейса в первую очередь необходимо организовать web-сервер. Предполагается, что web-сервер и сервер СУБД будут находиться в пределах одной локальной вычислительной сети, что обеспечит достаточно большую скорость обмена данными между ними.

На web-сервере хранятся сценарии, которые содержат логику обработки информационных запросов пользователей и отображения результатов в виде web-страниц. На схеме взаимодействия компонентов системы (рисунок 1) web-сайт обозначен как 'БД АСУ "Учебный процесс" для Web'.

Пользователи могут получить доступ к данным, хранящимся в информационной базе, через Интернет или локальную сеть университета с помощью обычного web-браузера.

В дальнейшем планируется расширение системы путем создания модулей для взаимодействия преподавателей и студентов и развертывания системы тестирования знаний, которая сможет учесть не только ответы тестируемых студентов, но и их успеваемость по дисциплинам, входящим в тестирование.

Таким образом, при данном варианте структуры информационной системы («тонкий» клиент) снимаются высокие требования к производительности рабочих станций пользователей и к пропускной способности канала связи между клиентом и сервером. При этом основная нагрузка ложится на web-сервер и канал связи между ним и сервером баз данных. Такой вариант организации системы наиболее пригоден для систем с большим количеством пользователей и различным аппаратным и программным обеспечением пользовательских рабочих станций.

Литература

1. Гринченков Д.В., Панфилов А.Н., и др. Сравнительный анализ систем автоматизации образовательного процесса / Теория, методы проектирования, программно техническая платформа корпоративных информационных систем: Материалы Междунар. науч.-практ. конф., г. Новочеркасск / Юж.-Рос. гос. техн. ун-т. – Новочеркасск: ЮРГТУ, 2003. – С. 39-43.
2. Барбаняга П.С., Панфилов А.Н., Лебедев В.Ю. Проект веб-сайта модуля управления учебным процессом ИАСУ ВУЗ ЮРГТУ (НПИ) / Сборник конкурсных работ Всероссийского смотра-конкурса научно-технического творчества студентов высших учебных заведений «Эврика-2009», г. Новочеркасск, декабрь 2009 г. / Мин-во образования и науки РФ, Юж.- Рос. гос. техн. ун-т. (НПИ). – Новочеркасск: Лик, 2010. – С. 6-7

Чмыхов А.А., аспирант кафедры ИИСТ, e-mail: chmikhov@gmail.com

Панфилов А.Н., к.т.н., доцент кафедры ИИСТ, e-mail: panfiloff@rambler.ru

Орлов С.Б., студент кафедры ИИСТ, e-mail: sorankit09@gmail.com

УДК 004

ФОРМАЛИЗОВАННЫЙ АНАЛИЗ ФУНКЦИОНАЛА ИНФОРМАЦИОННЫХ СИСТЕМ ПО ФОРМИРОВАНИЮ БУХГАЛТЕРСКОЙ ОТЧЕТНОСТИ В СООТВЕТСТВИИ С МСФО

Широбокова С.Н., Левшина А.С.

Южно-Российский государственный технический университет (НПИ), г. Новочеркасск

Представлен сравнительный анализ информационных систем по критерию функциональной полноты возможности формирования бухгалтерской отчетности в соответствии с международными стандартами финансовой отчетности.

Анализ по критерию функциональной полноты позволяет количественно сопоставить между собой информационные системы и оценить их соответствие требованиям пользователя. Сравним информационные системы по критерию функциональной полноты реализации возможности формирования бухгалтерской отчетности в соответствии с МСФО [1].

Пусть $Z = \{Z_i\} (i = 1, 2, \dots, 5)$ – множество сравниваемых информационных систем, реализующих формирование финансовой отчетности МСФО.

Z_1 – 1С:Управление производственным предприятием 8;

Z_2 – Инотек Бухгалтер Вестерн;

Z_3 – Система Галактика ERP;

Z_4 – ERP Монолит SQL;

Z_5 – Условная система, требования к которой отражают необходимый некоторому пользователю функционал.

Выделим укрупнено основные функции, которые должны выполнять система формирования отчетности в соответствии МСФО:

1) Формирование плана счетов МСФО. В системе должно быть несколько планов счетов. Один предназначен для учета операций в соответствии РСБУ, а другой план счетов – для отражения операций МСФО. В плане счетов для МСФО могут присутствовать счета, которых нет в российском плане счетов.

2) Мэппинг плана счетов представляет собой возможность настраивать таблицу соответствия синтетических (аналитических) разрезов между планами счетов.

3) Мультивалютный учет. В соответствии со стандартами учета отчетность по РСБУ формируется в рублях, по МСФО – в функциональной валюте, которая может отличаться от рублей.

4) Фиксация операций хозяйственной деятельности, отличных от РСБУ.

5) Отражение проводок, совершенных в предыдущем периоде РСБУ, но отраженных в новом периоде МСФО.

6) Функция формирование отчетности включает в себя формирование полного комплекта отчетности, соответствующего требованиям МСФО: отчет о финансовом положении, отчет о совокупном доходе, отчет о движении денежных средств, отчет об изменениях в капитале.

7) Консолидированный учет. По международным и российским правилам материнская компания обязана консолидировать все зарубежные и национальные предприятия концерна независимо от рода их деятельности.

8) Создание произвольных отчетов, так как МСФО не регламентируют четкую форму отчетов.

Тогда $R = \{R_j\} (j = 1, 2, \dots, 8)$ – это множество, составляющее словарь реализуемых системами $\{Z_i\}$ функций. Исходную информацию представим в виде таблицы $\{X_{ij}\}$ (табл. 1).

Таблица 1

Результаты оценки функциональной полноты систем Z_i

Наименование программной системы	Наименование выполняемой функции								
	R_1	R_2	R_3	R_4	R_5	R_6	R_7	R_8	$\sum_{j \in 8} R_j$
Z_1	1	1	1	1	1	1	1	0	7
Z_2	1	1	1	0	0	1	1	0	5
Z_3	1	1	1	1	1	1	0	0	6
Z_4	1	1	1	0	0	1	1	0	5
Z_5	1	1	1	1	1	1	1	1	8
$\sum_{i \in 5} R_i$	5	5	5	3	3	5	4	1	

Элементы таблицы $\{X_{ij}\}$ определяются по следующему правилу:

$$X_{ij} = \begin{cases} 1, & \text{если } j\text{-я функция реализуется } i\text{-ой системой;} \\ 0, & \text{если не выполняется.} \end{cases}$$

По данным табл. 1 вычислим матрицы $P^{(01)}$, G и H , где:

$P_{ik}^{(01)}$ – число функций, выполняемых системой Z_k , но не реализуемых, т.е. $P_{ik}^{(01)} = |Z_k / Z_i|$ – мощность разности множества $Z_k = \{X_{kj}\}$ и $Z_i = \{X_{ij}\}$.

$$P_{ik}^{(01)} = \begin{vmatrix} 0 & 0 & 0 & 0 & 1 \\ 2 & 0 & 2 & 0 & 3 \\ 1 & 1 & 0 & 1 & 2 \\ 2 & 0 & 2 & 0 & 3 \\ 0 & 0 & 0 & 0 & 0 \end{vmatrix}$$

Величина H_{ik} используется для оценки части функций, выполняемых системой Z_i , а также реализуемых в системе Z_k .

$$H_{ik} = P_{ik}^{(11)} / (P_{ik}^{(11)} + P_{ik}^{(10)}), \quad (0 \leq H_{ik} \leq 1),$$

где $P_{ik}^{(11)}$ – число функций, выполняемых и Z_i , и Z_k , т.е. $P_{ik}^{(11)} = |Z_i \cap Z_k|$ – мощность пересечения множества $Z_k = \{X_{kj}\}$ и $Z_i = \{X_{ij}\}$; $P_{ik}^{(10)}$ – число функций, выполняемых системой Z_i , не реализуемых системой Z_k , т.е. $P_{ik}^{(10)} = |Z_i / Z_k|$ – мощность разности множества $Z_k = \{X_{kj}\}$ и $Z_i = \{X_{ij}\}$.

$$H_{ik} = \begin{vmatrix} 1 & 0,714 & 0,857 & 0,714 & 1 \\ 1 & 1 & 0,8 & 1 & 1 \\ 1 & 0,666 & 1 & 0,666 & 1 \\ 1 & 1 & 0,8 & 1 & 1 \\ 0,875 & 0,625 & 0,75 & 0,625 & 1 \end{vmatrix}$$

Взаимосвязь между системами Z_i и Z_k оцениваются по значениям $P_{ik}^{(11)}$ и $G_{ik} = P_{ik}^{(11)} / P_{ik}^{(00)}$, ($0 \leq G_{ik} \leq 1$), где G_{ik} – мера подобия Жаккарда; $P_{ik}^{(00)} = |Z_i \cup Z_k|$ – мощность объединения Z_i и Z_k , т.е. $P_{ik}^{(00)} = P_{ik}^{(11)} + P_{ik}^{(10)} + P_{ik}^{(01)}$.

$$G_{ik} = \begin{vmatrix} 1 & 0,714 & 0,857 & 0,714 & 0,875 \\ 0,714 & 1 & 0,571 & 1 & 0,625 \\ 0,857 & 0,571 & 1 & 0,571 & 0,75 \\ 0,714 & 1 & 0,571 & 1 & 0,625 \\ 0,875 & 0,625 & 0,75 & 0,625 & 1 \end{vmatrix}$$

Выбирая различные пороговые значения ε элементов матриц $P^{(01)}$, G и H , построим логические матрицы поглощения $P_0^{(01)}$, G_0 и H_0 . Например, элементы матрицы $P_0^{(01)}$, получают следующим образом:

$$P_{ik}^{(01)} = \begin{cases} 1, \text{ если } P_{ik}^{(01)} \geq \varepsilon_p \text{ и } i \neq k; \\ 0, \text{ если } P_{ik}^{(01)} < \varepsilon_p \text{ или } i = k. \end{cases}$$

Для ранжирования систем по критерию функциональной полноты построим граф превосходства, используя в качестве порогового значения $\varepsilon_p = 1$. Тогда логические матрицы поглощения $P_0^{(01)}$ принимает следующий вид:

$$P_0^{(01)} = \begin{vmatrix} 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 \end{vmatrix}$$

Граф превосходства представлен на рис. 1. Он показывает, как и насколько выбранная и сравниваемая системы превосходят друг друга. Как видно из рис. 1, наименьшую функциональную полноту имеют системы 2 и 4 – их превосходят остальные системы (1, 3, 5).

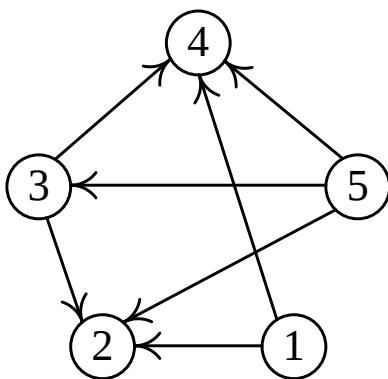


Рис. 1. Граф превосходства при $\varepsilon_p=1$

Задавая пороговые значения ε_g , получим матрицу поглощения G_0

$$G_{ik}^0 = \begin{cases} 1, \text{если } G_{ik} \geq \varepsilon_g \text{ и } i \neq k; \\ 0, \text{если } G_{ik} < \varepsilon_g \text{ или } i = k. \end{cases}$$

Используя в качестве порогового значения $\varepsilon_g = 0,8$, получим следующую матрицу поглощения.

$$G^0 = \begin{vmatrix} 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{vmatrix}$$

На рис. 2 приведен граф подобия, построенный по матрице G_0 при $\varepsilon_g = 0,8$:

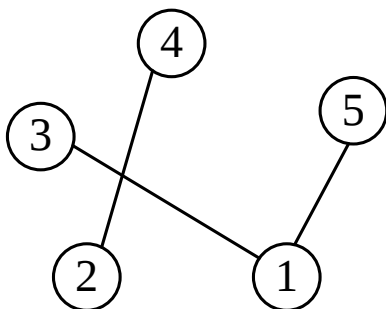


Рис. 2. Граф подобия при $\varepsilon_g = 0,8$

Задав , получим следующую матрицу G_0 :

$$G^0 = \begin{vmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{vmatrix}$$

На рис. 3 приведен граф подобия, построенный по матрице G_0 при $\varepsilon_g = 0,9$:

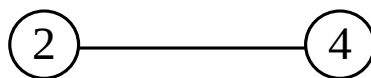


Рис. 3. Граф подобия при $\varepsilon_g = 0,9$

Как видно из рис. 2, группы схожих систем составляют системы 1,3 и 2,4. С помощью построенного графа пользователю наглядно видно, что если его заинтересовала система 1, то следует обратить внимание и на 3. Рис. 3 свидетельствует о том, что наибольшую степень схожести имеют системы 2 и 4.

Задав пороговое значение $\varepsilon_n = 0,9$, построим матрицу поглощения по следующему правилу:

$$H_{ik}^0 = \begin{cases} 1, & \text{если } H_{ik} \geq \varepsilon_n \text{ и } i \neq k; \\ 0, & \text{если } H_{ik} < \varepsilon_n \text{ или } i = k. \end{cases}$$

Выбрав пороговое значение $\varepsilon_n = 0,9$, получим следующую матрицу поглощения H^0 :

$$H^0 = \begin{vmatrix} 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{vmatrix}$$

По результатам матрицы H^0 построим граф поглощения (рис. 4):

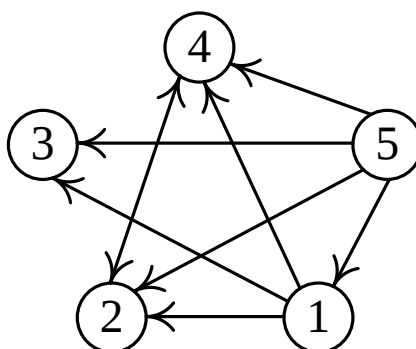


Рис. 4. Граф поглощения при $\varepsilon_n = 0,9$

Из рис. 4 видно, что системы 4 и 2 более чем на 90 % поглощаются системами 1 и 5.

Объектом данного исследования являются информационные системы бухгалтерского учета в соответствии с МСФО. Хотя в перечень анализируемых систем были включены комплексные системы автоматизации хозяйственной и финансовой деятельности предприятия, исследовалась лишь функциональная полнота реализации учета в соответствии с МСФО. В результате проведенного анализа можно отметить следующее: наиболее функционально полной системой является система 1. Системой 2 были показаны наиболее низкие результаты. Это объясняется отсутствием реализации некоторых основных функций в системе.

Литература

1. Хубаев Г.Н. Сравнение сложных программных систем по критерию функциональной полноты / Г.Н. Хубаев // Программные продукты и системы (SOFTWARE&SYSTEMS). – 1998. – № 2. – С. 6-9.

СОДЕРЖАНИЕ

Вороненко А.В., Щербакова Е.А. Информационно-аналитическая система отдела кадров коммерческого предприятия	3
Гавриков М.М., Синецкий Р.М. Алгоритм поиска наилучшей области определения при оптимальном синтезе и распознавании речевых образов	5
Горбатова Ю.А. Сравнительный анализ подходов к подготовке IT-специалистов .	10
Горбатова Ю.А. Обзор возможностей облачных технологий	15
Горбатова Ю.А. Отдельные аспекты методики построения системы защиты персональных данных	19
Горобец В.В. Оптимизация размещения фрагментов базы данных по узлам сети с облачной архитектурой по критерию минимума стоимости хранения	22
Горобец В.В. Оптимизация размещения фрагментов РБД в узлах сети с произвольной топологией в рамках облачной структуры по критерию минимума стоимости трафика	28
Горобец В.В. Модель надежности функционирования транзакционной системы, построенной на базе облачной архитектуры	35
Гринченков Д.В., Горбушин А.А., Горбушин Д.А. Алгоритм построения скоринговых моделей на основе дерева решений	50
Гринченков Д.В., Горбушин Д.А., Горбушин А.А. Программные продукты для анализа тональности текста, текущее состояние и реализуемые возможности	56
Гринченков Д.В., Горяинова О.Н. Актуальность разработки инструментального комплекса для формирования интегрированной среды управления образовательной деятельностью в системе среднего профессионального образования	68
Гринченков Д.В., Куций Д.Н. Базовые принципы разработки и пример реализации поискового робота	77
Гринченков Д.В., Шаменко А.А. Принципы построения интеллектуальной системы автоматизированного проектирования Web-приложений	84
Дущенко Т.К., Нечаев А.В. Эффективность информационных систем управления	95
Ковалевский Д.В., Широбокова С.Н. Медицинская интерактивная информационная система «MediaCare»: основные концепции, системные и технические решения	101
Ковалевский В.Н. Концепции и вопросы проектирования информационных систем абонентского обслуживания населения для предприятий водоснабжения и водоотведения	114

Костеникова Ю.С., Щербакова Е.А. Информационно-аналитическая система оценки бизнес-планов начинающих предпринимателей	125
Орловский Н.М. Графическая библиотека в составе автоматизированной системы планирования РС МКС	128
Орловский Н.М. Введение в процесс планирования полетов РС МКС	132
Погорелов А.С. Концепция построения системы поддержки принятия решений на основе понятия нечеткой онтологии	140
Погорелов А.С., Панфилов А.Н. О методах теории нечетких множеств в информационных системах	145
Сейтумерова Л.Б., Воробьев С.П. Реализация автоматизированной системы электронного документооборота для отдела внешних связей ООО «КЗ «Ростсельмаш»	147
Сейтумерова Л.Б., Воробьев С.П. Математические модели систем электронного документооборота	153
Сергеев Д.А. Структурные преобразования в расчетах надежности систем	156
Черноморова Т.С., Богучарская А.Н. Информационные системы в области испытаний вертолетных двигателей	161
Чмыхов А.А., Михайлов А.А. Способ повышения эффективности компьютерного тестирования	166
Чмыхов А.А., Панфилов А.Н., Орлов С.Б. Об одном подходе к построению разнородных систем в едином информационном пространстве	167
Широбокова С.Н., Левшина А.С. формализованный анализ функционала информационных систем по формированию бухгалтерской отчетности в соответствии с МСФО	170

Научное издание

**Теория, методы проектирования, программно-техническая
платформа корпоративных информационных систем**

Материалы X Международной научно-практической конференции

Подписано в печать 11.09.2012 г.

Формат 60х84 1/16. Усл. печ. л. 11,5. Тираж 100 экз. Заказ

Южно-Российский государственный технический университет
(Новочеркасский политехнический институт)

Отпечатано в ИД «Политехник»

Адрес университета:

346428, г. Новочеркасск, ул. Просвещения, 132, тел. 55-303