ISSN 1998-0663

№1(15)–2011

http://bijournal.hse.ru

БИЗНЕС-ИНФОРМАТИКА

МЕЖДИСЦИПЛИНАРНЫЙ НАУЧНО-ПРАКТИЧЕСКИЙ ЖУРНАЛ НИУ ВШЭ

BUSINESS INFORMATICS

Учредитель:
Национальный исследовательский университет «Высшая школа экономики»

Редакционная коллегия

Абдульраб А. (Франция)
Авдошин С.М.
Алескеров Ф.Т.
Белов В.В.
Грибов А.Ю.
Громов А.И.
Гюнтер Х. (Германия)
Ильин Н.И.
Калягин В.А.
Каменнова М.С.
Козырев О.Р.
Кузнецов С.О.
Мальцева С.В.
Миркин Б.Г. (Великобритания)
Моттль В.В.
Мулазани М. (Италия)
Пальчунов Д.Е.
Силантьев А.Ю.
Таратухин В.В.
Терзани С. (Италия)

В ЭТОМ НОМЕРЕ:

СИТУАЦИОННЫЕ ТЕХНОЛОГИИ УПРАВЛЕНИЯ

«ТЕОРИЯ СЛОЖНОСТИ»

КОМПОНЕНТНЫЕ АРХИТЕКТУРЫ

СЕМАНТИЧЕСКОЕ ПРЕОБРАЗОВАНИЕ ЗАПРОСОВ

В соответствии с решением президиума ВАК РФ журнал «Бизнес-информатика» с 19.02.2010 включён в Перечень ведущих рецензируемых научных журналов и изданий, в которых должны быть опубликованы основные научные результаты диссертаций на соискание ученых степеней кандидата и доктора наук.

БИЗНЕС-
ИНФОРМАТИКА
№1(15)–2011

СОДЕРЖАНИЕ

Бизнес-интеллект и принятие решений

И.С. Вахмянин, Н.И. Ильин, Е.В. Новикова
Разработка модели алгоритма управления информационными потоками в ситуационных центрах органов государственной власти 3

Математические методы и алгоритмы решения задач бизнес-информатики

В.А. Кохов, В.В. Кохов
Метод решения задачи различения орграфов на основе сложности..... 11

Информационные системы и технологии в бизнесе

Е.М. Гринкруг, А.Р. Шакуров
Компонентная архитектура с определением типов во время исполнения 24

Н.Н. Лычкина, А.Р. Идиатуллин
Инструментальная реализация архитектурных моделей предприятия на основе онтологий 31

Моделирование и анализ бизнес процессов

С.М. Авдошин, Е.Ю. Песоцкая
Информационные технологии для управления финансовыми рисками 42

Т.К. Богданова, Ю.А. Алексеева
Прогнозирование вероятности банкротства предприятий с учетом изменения финансовых показателей в динамике..... 50

Анализ данных и интеллектуальные системы

А.В. Кириллов, В.А. Фомичев
О новом подходе к семантическому преобразованию естественно-языковых запросов поисковых систем 61

Annotations 69

БИЗНЕС-ИНФОРМАТИКА
№1(15)–2011

Междисциплинарный
научно-практический журнал
НИУ ВШЭ

Журнал рекомендован ВАК
для научных публикаций

Подписной индекс издания
в каталоге агентства
«Роспечать» –72315

Учредитель:
Национальный
исследовательский университет
«Высшая школа экономики»
Выходит 4 раза в год.

Научный редактор
Лычкина Н.Н.

Технический редактор
Осипов В.И.

Дизайн обложки
Борисова С.Н.

Компьютерная вёрстка
Богданович О.А.

Администратор веб-сайта
Проценко Д.С.

Адрес редакции:
105679, г. Москва,
ул. Кирпичная, д. 33.
Тел.+7 (495) 771-32-38,
e-mail: bijournal@hse.ru

За точность приведённых сведений
и содержание данных,
не подлежащих открытой публикации,
несут ответственность авторы

При перепечатке ссылка на журнал
«Бизнес-информатика» обязательна

Тираж 500 экз.

Отпечатано в типографии НИУ ВШЭ
г. Москва, Кочновский проезд, 3.

© Национальный
исследовательский университет
«Высшая школа экономики»

РАЗРАБОТКА МОДЕЛИ АЛГОРИТМА УПРАВЛЕНИЯ ИНФОРМАЦИОННЫМИ ПОТОКАМИ В СИТУАЦИОННЫХ ЦЕНТРАХ ОРГАНОВ ГОСУДАРСТВЕННОЙ ВЛАСТИ

И.С. Вахмянин,
руководитель отдела разработки и внедрения программного обеспечения ЗАО «Полимедиа», аспирант Российской академии государственной службы при Президенте РФ,
e-mail: ivan@polymedia.ru.

Н.И. Ильин,
доктор технических наук, профессор, заместитель начальника Управления информационных технологий спецсвязи ФСО РФ.

Е.В. Новикова,
кандидат химических наук,
Генеральный директор ЗАО «Полимедиа»,
докторант Российской академии государственной службы при Президенте РФ,
e-mail: elena@polymedia.ru.

Адрес: г. Москва, ул. Кржижановского, д. 29, корп. 1, ЗАО «Полимедиа».

Разработана математическая модель информационных потоков в ситуационном центре (СЦ) и на ее основе построен алгоритм рационального управления информационными потоками. На базе разработанного алгоритма реализовано программно-аппаратное решение «Визуализация информации на системе распределенных дисплеев» (ВИРД), которое внедрено в ситуационном центре Пензенской области.

Ключевые слова: визуализация, ситуационный центр (СЦ), модель информационного потока, информационно-аналитическая система, информационно-аналитическая система, информационный поток, визуализация информации на системе распределенных дисплеев (ВИРД).

Введение

Система ситуационных центров (СЦ) обеспечивает повышение эффективности, оперативности и качества принятия управленческих решений, а также контроля исполнения решений по различным направлениям деятельности органов государственной власти и органов местного самоуправления. Сегодня в России насчитывается около 30 реализованных СЦ, и новые центры продолжают активно создаваться [1]. Известно [2,3] что до 80 % воспринимаемой информации человек получает по зрительному каналу, поэтому подсистема отображения информации является ключевой для СЦ и центров управления, которые предназначены для поддержки принятия решения в условиях ограниченного времени или в условиях чрезвычайной ситуации. При этом информация, поступающая в СЦ носит «лавинообразный», неформализованный и неравномерный характер; информация поступает от внешних и внутренних, программных и аппаратных информационных источников (оперативные донесения, ленты новостей, транскрипты ТВ-программ, Интернет-сайты). В таких условиях ключевыми факторами эффективной работы СЦ является время, затраченное на восприятие информации лицом, принимающим решения, и другими участниками совещания [4]. Для этого организация вывод информации на систему экранов СЦ должна быть достаточно простой и оперативной.

Долгое время развитие ситуационных центров и систем управления выводом информации сдерживалось несовершенством средств обработки и визуализации информации. В частности, источник информации был, как правило, жестко привязан к средству отображения информации [5]. То есть конфигурация ситуационного центра была статической и описывалась парами «источник информации» — «средство отображения». Современный СЦ представляет собой программно-аппаратный комплекс, в котором множество источников информации (программных и аппаратных) выводится на множество дисплеев, причем, этим процессом можно управлять в реальном времени.

Для подготовки совещания требуется разработать сценарий проведения совещания, определить источники информации, способы их представления и последовательность представления информации, а также описать последовательность представления информации в виде сцен. Определенную сложность будет представлять движение по сценарию и переход от одной сцены к другой, вследствие того, что каждый информационный поток представлен своим источником, и вывод информации осуществляется в многооконном режиме на несколько экранов. Выбор соответствующего источника и расположение его на экране занимает временной ресурс, который является очень важным для эффективного функционирования СЦ. Система коммутации оборудования СЦ является сложной и накладывает определенные ограничения на вывод информационных источников на экран коллективного пользования (ЭКП). Если оператор осуществляет коммутацию «вручную» (простым перебором), то существует риск, что не все источники информации будут выведены на ЭКП. Задачу вывода изображения на экран большой информационной емкости можно решать на программном или аппаратном уровне. В работе [6] такая задача решается с помощью системы многопользовательских контроллеров, однако такая система является достаточно громоздкой, сложной в управлении (при наращивании числа источников информации) и не позволяет применить сценарный подход к организации совещаний. Применение систем управления, разработанных фирмами Crestron [7] и AMX [8] дает возможность легко управлять выводом изображения от аппаратных источников. Однако, производительность данных систем относительно невысока, что делает невозможным программирование действительно сложной логики с учётом требований режима реального времени. Кроме того, системы интегрированного управления изначально ориентированы на создание несложных сценариев управления оборудованием, что затрудняет их взаимодействие с программными источниками информации и, тем более, информационными системами. Напротив, программное обеспечение Authorware (www.macromedia.com) предоставляет пользователю возможность создавать разветвленные сценарии с достаточно сложной логикой связей между слайдами. Широкий набор средств для создания сложных сценариев показа сделал данный продукт часто используемым в ситуационных центрах. Основным недостатком Authorware является невозможность работы с многодисплейными конфигурациями, и управления оборудованием. Также Adobe Authorware предоставляет не очень широкие возможности по отображению актив-

ных (не статических) источников, таких как визуализация витрин данных, «живые» отчеты и т.п. Для решения вышеуказанных проблем необходима разработка специального программного обеспечения (ПО), предназначенного для управления информационными потоками и оптимизации представления информации, выводимой на экран коллективного пользования. Такое ПО должно обеспечивать решение следующих задач:

✦ подготовка сценариев отображения информации с использованием различных типов источников (как аппаратных, так и программных);

✦ легкое управление выводом информации с любого источника (программного или аппаратного) на любой из экранов СЦ;

✦ управление совещанием (как для операторов, так и для ЛПР) с возможностью оперативного переключения между сценариями и управления любым источником;

✦ визуализация витрин данных, находящихся в информационно-аналитических системах используемых в СЦ.

Модель информационных потоков

Необходимость создания средств и методов рационального управления информационными потоками продиктовала потребность в создании программно-аппаратного комплекса, опирающегося на формальную математическую модель информационных потоков в СЦ. Далее совокупность информационных сообщений, характеризующих состояние обсуждаемых вопросов, поступающих в СЦ в различных форматах (например, текстовые документы, табличные данные, картографическая информация, видеофайлы) и от различных источников, через определенные промежутки времени, будем называть **информационным потоком**.

Рассмотрим пример упрощенной схемы соединений ситуационного центра *(рис. 1)*. Предположим, что оператору нужно вывести отчет из информационно-аналитической системы 1 (ИАС1) на ЖК-панель (пассивный дисплей), а отчет из ИАС 2 на видеостену. Если оператор будет произ-

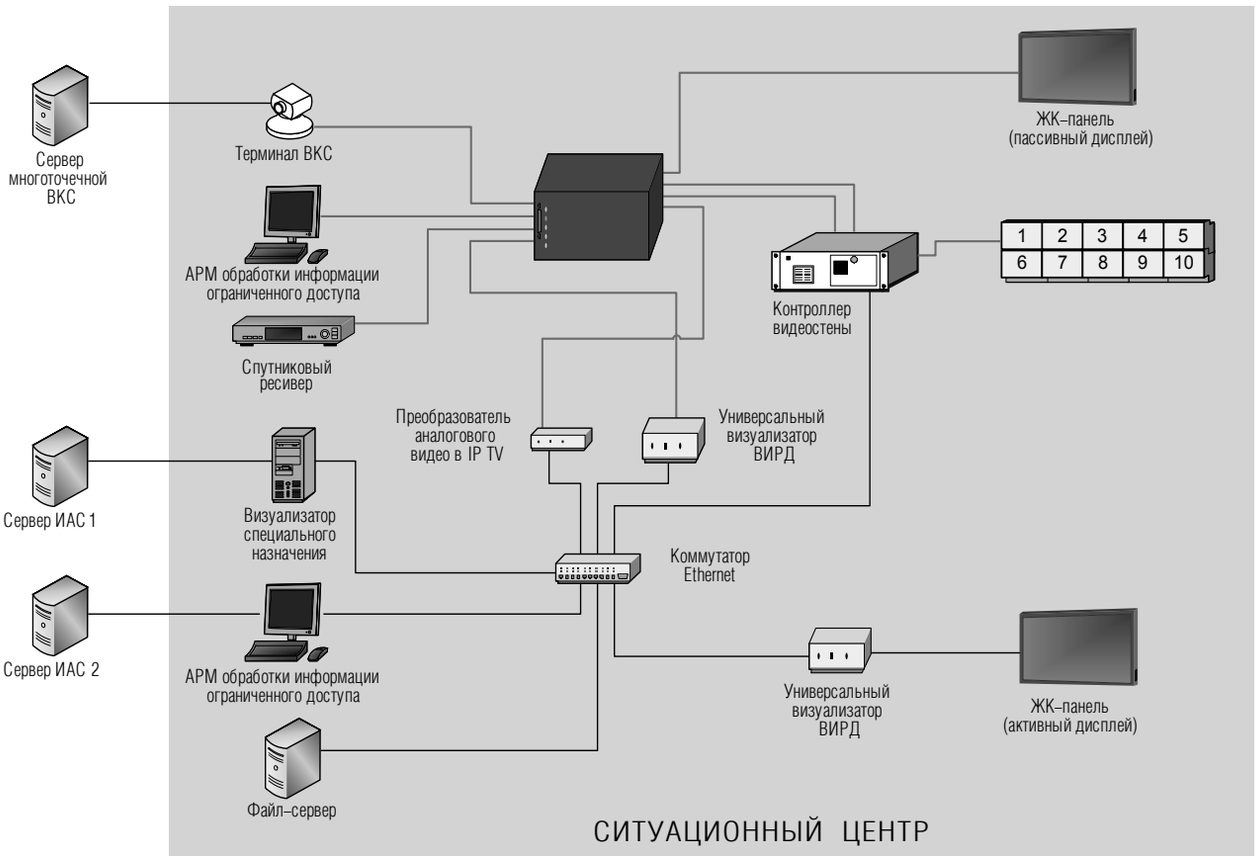


Рис. 1. Схема соединений ситуационного центра.

водить коммутацию вручную, он может вывести отчет из ИАС 2 на видеостену следующим образом: «АРМ обработки информации ограниченного доступа» ⇒ «Коммутатор Ethernet» ⇒ «Универсальный визуализатор» ⇒ «Матричный коммутатор» ⇒ «Контроллер видеостены». Однако в этом случае вывести на ЖК-панель отчет из ИАС 1 будет уже невозможно, так как иного способа сделать это, кроме как используя универсальный визуализатор, нет, а он уже занят отчетом из ИАС 2. Если бы вывод этого отчета осуществлялся напрямую с коммутатора Ethernet на контроллер видеостены, то такой проблемы бы не возникло. Можно представить, насколько более сложно принимать оператору подобные решения в реальном ситуационном центре, где число объектов схемы коммутации может достигать значения 10 и более.

Имеется ряд систем управления информационным потоком (ИП) на техническом уровне, однако они сводятся, в основном, к аппаратному либо программному управлению ИП и не решают задачи коммутации множества источников на множество экранов.

Для решения описанной задачи потребовалось разработать непротиворечивую и полную математическую модель, описывающую информационные потоки ситуационного центра. Основной целью создания такой модели является сокрытие вариативности используемых в СЦ ресурсов и подготовка инструментария для формального описания состояния СЦ и его информационных потоков. Для достижения этих целей была выбрана модель в виде информационной сети, описанной графом. Данный выбор обоснован тем, что в этом случае к решению задачи можно применить существующие алгоритмы и математические методы теории графов.

Представим информационную сеть СЦ в виде взвешенного ориентированного графа. Вершинами графа являются устройства СЦ. Источники информации имеют только исходящие ребра, средства отображения – только входящие, средства преобразования и коммутации – обоих типов. При построении данной модели используется терминология задачи о максимальном потоке в ориентированном графе [9].

Назовем **информационной сетью** СЦ ориентированный граф $G = (V, E)$, каждому ребру $(u, v) \in E$ которого поставлено в соответствие число $c(u,v) \geq 0$, называемое пропускной способностью ребра. В графе выделим две группы вершин – истоки S (источники информации) и истоки T (средства отображения информации). **Путем информационного потока** назовем упорядоченную последовательность вершин

$$r = (v_1, v_I, \ldots v_k),\ v_i \in V,\ v_I \in S,\ v_k \in T,\ i=\overline{1,k},$$

причем любые две последовательные вершины соединены хотя бы одним ребром из множества E. При этом пару $p = (v_1, v_k)$ будем называть **информационным соответствием**.

Таким образом, состояние информационных потоков ситуационного центра можно описать совокупностью информационных соответствий. При этом для каждого информационного соответствия будет существовать реализующий его путь информационного потока.

Используя данную модель, можно изобразить схему соединений СЦ (*рис. 1*), представленную выше, в следующем виде (*рис. 2*):

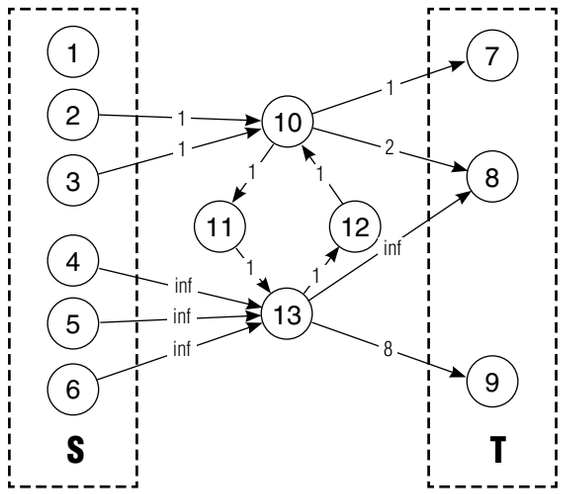


Рис. 2. Модель информационных ресурсов СЦ.

При данном построении были использованы следующие обозначения вершин:

Источники (S): u_1 — терминал ВКС, u_2 — АРМ обработки информации ограниченного доступа, u_3 — спутниковый ресивер, u_4 — визуализатор специального назначения, u_5 — АРМ обработки аналитической информации, u_6 — файл-сервер.

Дисплеи (T): u_7 — ЖК-панель, u_8 — контроллер видеостены (видеостена), u_9 — визуализатор информации на системе распределенных дисплеев (ЖК-панель).

Коммутационные ресурсы: u_{10} — матричный

коммутатор, u_{11} — преобразователь аналогового видео в IPTV, u_{12} — визуализатор информации на системе распределенных дисплеев, u_{13} — Ethernet коммутатор.

Рассмотрим пример распределения информационных потоков в данной модели. Пусть требуется отобразить спутниковое видео на пассивной ЖК-панели, а на видеостену в трех окнах — ВКС, отчет с ИАС 1 и видеофайл с файл-сервера. В терминах модели эту задачу можно описать четырьмя информационными соответствиями (парами вершин из *S, T* соответственно):

$$p_1 = (u_1, u_8), \quad p_2 = (u_3, u_8), \quad p_3 = (u_4, u_8), \quad p_4 = (u_6, u_7).$$

Рассмотрим возможные пути информационных потоков. Например, допустим, что для первого, второго и третьего информационного соответствия реализованы следующие пути информационных потоков:

$$p_1 \to r_1 = (u_1, u_{10}, u_8)$$
$$p_2 \to r_2 = (u_3, u_{10}, u_{11}, u_{13}, u_8),$$
$$p_3 \to r_3 = (u_4, u_{13}, u_{12}, u_{10}, u_8).$$

В графическом виде описанное распределение изображено на *рис. 3*.

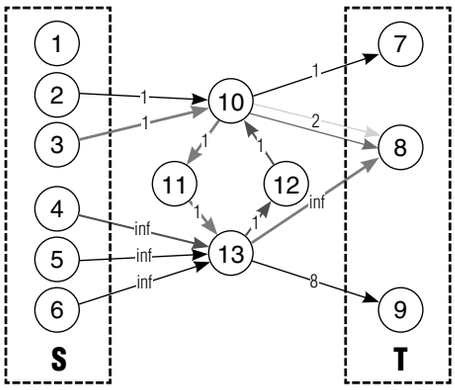
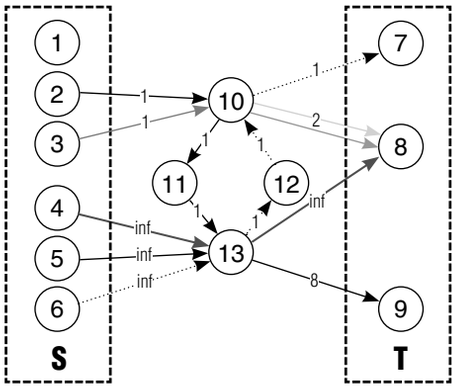


Рис. 3. Пути информационных потоков (вариант 1).

Очевидно, что при данном распределении невозможно реализовать четвертое информационное соответствие, т.е. единственный возможный путь из вершины u_6 в вершину u_7 проходит через ребро (u_{13}, u_{12}), пропускная способность которого исчерпана путем r_3. Соответственно, реализовать четвертое информационное соответствие не представляется возможным. Однако это не означает, что не существует другого варианта распределения информационных потоков, при котором возможно реализовать все четыре соответствия. Ниже представлен альтернативный способ распределения информационных потоков (*рис. 4*).

$$p_1 \to r_1 = (u_1, u_{10}, u_8)$$



$$p_2 \to r_2 = (u_3, u_{10}, u_8),$$
$$p_3 \to r_3 = (u_4, u_{13}, u_8).$$

Рис. 4. Пути информационных потоков (вариант 2).

Алгоритм управления информационными потоками

Поставленные в предыдущем разделе задачи могут быть решены как с помощью каких-либо специальных алгоритмов, так и с помощью простого перебора. Однако решение данных задач должно производиться в автоматическом режиме при каждом действии, связанном с изменением конфигурации СЦ (до 10 раз в минуту), время для получения ответа не должно превышать 1-5 секунд. С учетом экспоненциальной сложности полного перебора вариантов, было необходимо разработать алгоритм решения поставленной задачи распределения. Общее описание разработанного алгоритма приведено ниже.

Итак, входом для алгоритма будет информационная сеть СЦ $G = (V, E)$ и набор информационных соответствий $r_1, r_2, \ldots r_m$. Выход: набор путей информационных потоков $p_1, p_2, \ldots p_m$, реализующих заданные информационные соответствия, или сообщение о невозможности реализовать все из них. Общую схему алгоритма можно описать следующим образом:

1. Для каждого источника информации $s \in S$, такого что $\exists r, t : r = (s, t)$, находим все пути, ведущие из s в t такие, что каждое ребро графа встречается в пути не более одного раза (если такого ограничения не поставить, то таких путей могло бы быть бесконечно много при наличии циклов в графе).

2. Каждому ребру ставим в соответствие n чисел $k_1, k_2, \ldots k_m$, каждое из которых представляет количество проходящих через ребро возможных путей информационных потоков соответствующего типа.

3. Для каждого ребра, через которое проходят возможные пути информационных потоков более чем одного типа, поочередно пробуем удалить возможные пути разных типов.

4. При каждой попытке удаления путей выполняется проверка: существуют ли еще информационные потоки данного типа, кроме удаляемых. Если существуют, удаляем потоки и на ребрах, через которые они проходили, соответствующим образом уменьшаем число k. Если нет, переходим к следующему типу. Если не удается удалить информационные потоки с ребра, кроме одного, это означает, что данный набор пар невозможно реализовать на данной информационной сети. Если удается — переходим к следующему ребру.

5. Критерием положительной остановки алгоритма является отсутствие ребер, через которые проходит более одного возможного информационного потока (этот критерий аналогичен условию останова алгоритма, реализующего метод Форда-Фалкерсона [9] для решения задачи о максимальном потоке и минимальном разрезе).

Внедрение

На основе вышеописанных алгоритмов разработано ПО «ВИРД» (Визуализация информации на системе распределенных дисплеев). Основным отличием ПО «ВИРД» от большинства существующих систем является максимально простой и удобный пользовательский интерфейс, позволяющий пользователю без специальных технических знаний и понимания природы и свойств источников информации и средств отображения эффективно управлять процессом совещания и выводом необходимой информации. Этот подход позволил успешно интегрировать в одном продукте возможности работы с любыми источниками информации, будь то информационно-аналитическая система, обычный текстовый документ или, например система видеоконференцсвязи (ВКС). В результате система позволяет пользователю выводить информацию с них на любые имеющиеся средства отображения, причем не только находящиеся в одном помещении с СЦ. «ВИРД» использует сценарно-ориентированный подход: сценарий совещания описывается в «ВИРД» как нелинейная последовательность отдельных сцен, каждая из которых включает взаимное расположение графических окон от источников на ЭКП, а также состояние всего оборудования комплекса в каждый момент времени. Благодаря полной интеграции продукта с Microsoft Power Point, граф сцен сценария может быть получен автоматически из уже созданной презентации, что, несомненно, уменьшает время на адаптацию пользователей к новому продукту. ВИРД имеет модульную архитектуру.

Одним из примеров успешных внедрений «ВИРД» является Ситуационный центр Губернатора Пензенской области (*рис. 5*). В настоящее время СЦ работает в двух основных режимах: режиме «Планового совещания» и режиме «Аналитической телеконференции». В режиме «Планового совещания» материалы собираются аналитиком СЦ как с помощью специализированных ИАС, так и вручную с использованием универсальных средств связи. На АРМ аналитика производится обработка данных, их агрегирование и преобразование в вид, пригодный для визуализации. Полученные в результате этой работы файлы (графики, карты, схемы, отчеты) отправляются через файл-сервер на АРМ оператора визуализации. Оператор, используя ПО «ВИРД», создает сценарий совещания. Данный сценарий состоит из ряда сцен, каждая из которых содержит многооконные раскладки информационных образов на средствах отображения коллективного пользования (видеостена, мониторы на столах) и состояние оборудования. Последовательность сцен в сценарии определяется повесткой совещания, но может и содержать ветвления, а также заготовки для обсуждения вновь возникших во время совещания вопросов.

Во время совещания, оператор (или непосредственно ЛПР) управляет ходом сценария (переходом со сцены на сцену, а также информационными образами источников, представленных на экране). Важной особенностью данного технологического подхода к визуализации является то, что во время проведения совещания для создания информационных образов не используются АРМы операторов или аналитиков — визуализация происходит непо-

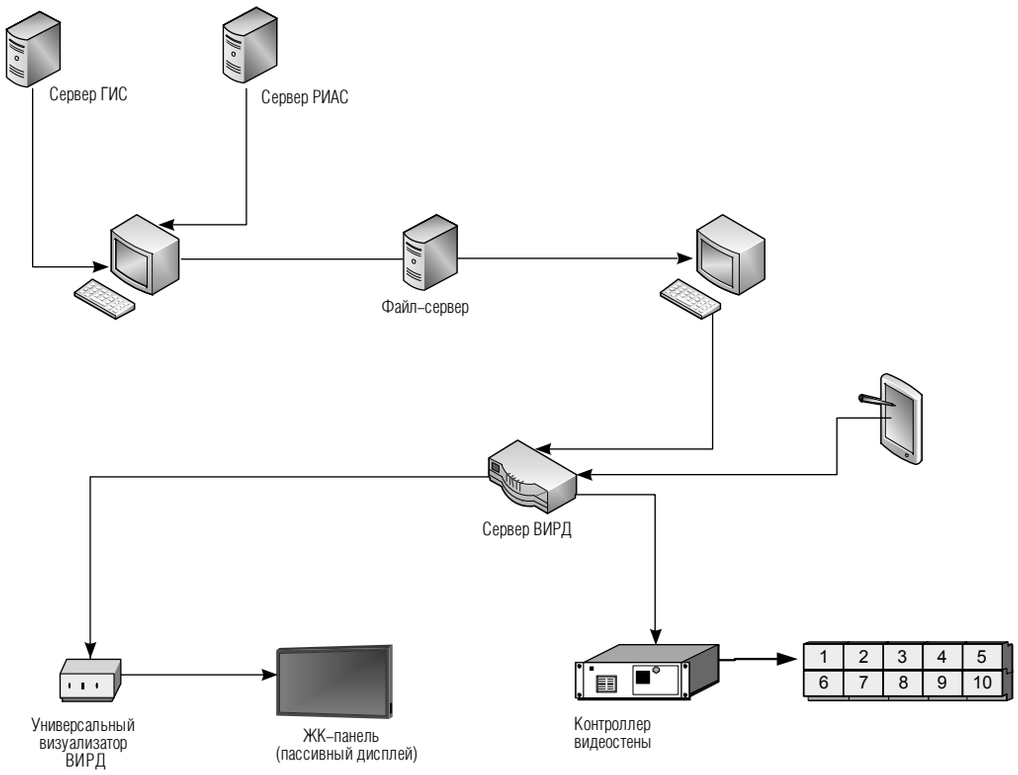


Рис. 5. Схема работы «ВИРД» в СЦ Администрации Пензенской области.

средственно на дисплеях (например, на видеостене). Это позволяет отображать сцены с произвольным количеством информационных образов при наличии всего двух АРМ — оператора и аналитика. Причем, поскольку АРМ аналитика также свободен, аналитик может использовать его во время совещания для поиска и подготовки какой-либо оперативно потребовавшейся дополнительной информации.

В режиме «Аналитической телеконференции» совещания с главами муниципальных образований проходят одновременно для 30 удалённых муниципальных образований. Ранее на удалённых абонентских пунктах отсутствовала возможность видеть материалы, документы и подготовленную аналитическую информацию, отображаемые на основных экранах в СЦ во время совещаний. Это приводило к тому, что для удалённых абонентов отсутствовала возможность полноценно работать по обсуждаемым вопросам повестки совещания. В связи с этим руководством Управления информатизации Пензенской области было сделано предложение, реализовать на базе ПО «ВИРД» функцию обеспечения трансляции блока информационно-аналитической поддержки на оборудование удаленных абонентских пунктов.

Сложность решения поставленной задачи состояла в том, что наряду с низкой пропускной способностью каналов связи (2 Мбит/с) экраны в СЦ и в муниципальных образованиях имеют разную информационную емкость (в СЦ видеостена и ноутбуки, а в муниципальных образованиях – проекторы или ЖК-дисплеи).

Решением данной задачи стало обеспечение автоматического режима загрузки на визуализаторы блока информационной поддержки в ходе технического сеанса ВКС с последующим одновременным выводом на дисплеи удаленных абонентов и управление показом этой информации во время рабочего сеанса. Для реализации данного решения в условиях описанных ограничений было внедрено синхронизированное управление отображением информации под управлением «ВИРД» в муниципальных образованиях. Это позволило оператору СЦ во время подготовки сценария располагать информационные образы не только на локальных дисплеях, но и на удаленных абонентских пунктах, учитывая при этом их информационную емкость. Соответствен-

по сети передаются лишь управляющие сигналы, что позволило экономить пропускную способность канала для видео- и аудиопотоков.

Программное обеспечение «ВИРД» было также внедрено в СЦ Губернатора Сахалинской области. ПО «ВИРД» решает для СЦ губернатора целый ряд задач: разработка сценариев отображения информации, оперативное управление сценариями и режимами отображения, а также демонстрация различных аналитических и медиа-данных в форме, удобной для восприятия. Особенностью данного внедрения является использование «ВИРД» для одновременного управления отображением информации из открытого и закрытого контуров без нарушения режима безопасности. Это было достигнуто при помощи разработки специальной системы коммутации.

Заключение

Таким образом, для оптимизации работы ситуационного центра разработана математическая модель информационных потоков СЦ и на ее основе построен алгоритм оптимального управления информационными потоками. На базе разработанного алгоритма реализовано программно-аппаратное решение ВИРД, которое внедрено в ситуационном центре Пензенской области. Практика внедрения ПО «ВИРД» показала повышение эффективности использования программно-технических средств СЦ за счет автоматизации и оптимизации процессов подготовки и управления информационными потоками, разработки сценариев отображения информации, оперативного управления сценариями и режимами отображения, а также демонстрации различных аналитических и медиаданных в форме, удобной для восприятия и работы ЛПР. ■

Литература

1. Ильин Н.И., Демидов Н.Н., Попович П.Н. Развитие систем специального информационного обеспечения государственного управления. – М.: Медиа Пресс, 2009.
2. Середа Г.К., Бочарова С.П., Репкина Г.В. Инженерная психология / Под ред. Г.К. Середы. – Киев: Вища школа, 1976. – 308 с.
3. Кучерявый А.А. Бортовые информационные системы / Под ред. В.А. Мишина и Г.И. Клюева. -2-е изд. перераб. и доп. – Ульяновск: УлГТУ, 2004. – 504 с.
4. Новикова Е.В. Создание ситуационных центров на базе аудиовизуальных и информационно-коммуникационных технологий // Материалы конф. «Ситуационные центры: модели, технологии, опыт практической реализации». – М.: РАГС, 2006. - 25с.
5. Ишеев И.А. Технические подсистемы и специализированное ПО для эффективной работы современного СЦ. //Материалы конф./ «Ситуационные центры и информационно-аналитические средства поддержки принятия решений». – М.: РАГС, 2009.
6. Мезенцев А.А., Павлов В.М., Шарнин А.В. Управление, вычислительная техника и информатика/ Двухуровневая компьютерная видеосистема с большой панелью коллективного или группового пользования
7. Control Systems for Home Automation, Campus & Building Control by Crestron Electronics, 2010. URL: http://www.crestron.com/default.asp (дата обращения: 10.09.2010)
8. Partridge Charles W. (Wylie, TX, US), Barber Ronald W. (Plano, TX, US), Lee Mark R. (Richardson, TX, US), Holub Douglas R. (Irving, TX, US), System and method for multimedia display, United States, AMX LLC (Richardson, TX, US), 2008.
9. Кормен Т., Ч. Лейзерсон, и Р. Ривест. Алгоритмы: построение и анализ.– М.: МЦНМО, 2001. -535с.

МЕТОД РЕШЕНИЯ ЗАДАЧИ РАЗЛИЧЕНИЯ ОРГРАФОВ НА ОСНОВЕ СЛОЖНОСТИ

В.А. Кохов,
кандидат технических наук, доцент кафедры Высшей математики на факультете экономики Национального исследовательского университета «Высшая школа экономики».

В.В. Кохов,
студент кафедры «Прикладная математика» Московского энергетического института,
e-mail: viktorkokhov@rambler.ru.
Адрес: г. Москва, ул. Красноказарменная, д. 14.

Предложен метод решения задачи различения орграфов. Основу метода определяет матричная модель сложности, которая учитывает количественные и качественные характеристики фрагментов орграфа. Модель впервые позволяет оценивать значимость каждого фрагмента орграфа в его общей сложности. Приведены результаты решения задач различения и определения сходства для орграфов.

Ключевые слова: орграф, сложность орграфа, проблема различения орграфов, модель сложности, сходство орграфов, расположение фрагментов.

1. Введение

Сложность системы, будучи ключевым понятием системологии, требует глубоких и интенсивных исследований и ориентирована на развитие подходов к анализу сходства систем.

Сложность и сходство структурированных объектов (корпоративных социальных сетей; семантических сетей, абстрактных графов, графов синтаксической структуры предложений и др.) наиболее востребована в таких областях, как системы искусственного интеллекта, информационно-поисковые системы структурной информации (семантического *web*-поиска текстовых документов) и др. [1-3].

Приведем краткий обзор по построению моделей сложности графов. Одна из первых моделей была связана с построением теоретико-информационных индексов [4,5]. Пусть имеется граф $G=(V, E)$, где $|V|=p$, $|E|=q$ и множество его вершин V, разбивается на k классов эквивалентности V_i, $i=1..k$. В качестве отношения эквивалентности может выступать принадлежность вершин к одной орбите группы автоморфизмов графа. Тогда *информационное содержание* (количество информации), приходящееся на одну вершину, будет определяться следующей формулой:

$$IIC(G/V) = -\sum_{i=1}^{k} C_i \log_2 C_i,$$

где $C_i=|V_i|/|V|$. *Полное информационное содержание* графа, рассчитанное на основе V, будет $ПИС(G/V)=|V|ИС(G/V)$. $ПИС(G/V)$ рассматривается как интегральная количественная мера его *структурной* сложности (*неоднородности*). На основе информационного содержания графа может быть построено множество различных индексов сложности, если вместо множества вершин анализировать другие множества однотипных фрагментов, например цепей с длиной два (P_2). Из [6] известен *орбитальный информационный индекс* графа в базисе связок $\eta(P_2)$:

$$ОИИ(G/P_2) = 2\eta \log_2 \eta - \sum_{i=1}^{k} \eta_i \log_2 \eta_i ,$$

где k – число орбит группы расположения η в G, η_i – количество связок в i-ой орбите. По сути, это полное информационное содержание графа, рассчитанное на основе связок, плюс «$\eta \log_2 \eta$» – информация о размере графа (число связок).

В [7] для определения сложности графа G введена функция

$$CM(G) = \frac{pq}{p+q} \sum_{i<j} \gamma(v_i, v_j),$$

где $\gamma(v_i,v_j)$ – число простых цепей из вершины v_i в вершину v_j. Указано, что эта функция соответствует интуитивному понятию *сложности графа*. В [8] предложена функция, значение которой, равно числу *остовных деревьев (каркасов)* графа. В качестве функции сложности в [9] используется число *деревьев*, изоморфно вкладываемых произвольным образом в граф G.

В [10] рассматривается топологическая сложность графовых моделей систем (ГМС), причем в наиболее общем и системном виде. Выделены девять требований, предъявляемых к моделям (функциям) сложности систем. В [11] приведен обзор подходов к определению структурной сложности графов и выделены три этапа в построении моделей: (1) построение индексов сложности, то есть числовых инвариантов графов, отражающих одну из сторон сложности, например сложность в терминах *цепных, циклических, древесных* и др. характеристик; (2) построение моделей сложности, основанных на мультииндексах, отражающих несколько сторон комплексного понятия «сложность»; (3) построение концептуальных и математических моделей, включающих многоуровневое отражение сложности, с привлечением достижений теории систем, теории графов, теории информации, топологии и др. Показано, что к перспективным направлениям построения моделей сложности ГМС *четвертого поколения* следует отнести такие *матричные модели*, которые учитывают три уровня характеризации в единой модели: (1) сложность ГМС как единого целого; (2) сложность фрагментов ГМС; (3) сложность расположения фрагментов ГМС в ее топологии с характеризацией вкладов в общую сложность.

Ниже предлагается матричная модель сложности орграфов, позволяющая:

✦ учитывать как количественные, так и качественные характеристики систем, представленных орграфами;

✦ обобщить индексы симметрии расположения «связок» до индексов симметрии расположения любых фрагментов орграфа и вычислять их значения все с большей и большей точностью;

✦ проводить иерархический анализ сложности орграфов с уточнением результатов на каждом следующем уровне иерархии;

✦ определять вклад любого фрагмента в общую сложность;

✦ все более и более точно решать задачи различения орграфов;

✦ все более и более точно решать задачи различения расположения фрагментов в орграфах;

✦ формировать и исследовать новые виды отношений эквивалентности орграфов.

2. Основные определения

Пусть $G=(V,E)$ орграф. Два орграфа $G=(V,E)$ и $G^*=(V^*,E^*)$ изоморфны ($G\approx G^*$), если

$$\exists\varphi: V \leftrightarrow V^* \text{ и } \forall u,\ v \in V[(u,v) \in E \leftrightarrow (\varphi(u),\varphi(v)) \in E^*],$$
$$\text{где } \varphi(u),\ \varphi(v) \in V^*.$$

Орграф $G^*=(V^*,E^*)$ изоморфно вкладывается в орграф $G=(V,E)$ как *порожденный подграф* ($G^* \subseteq {}^S G$), (подграф ($G^* \subset G$)), если в G есть порожденный подграф (подграф) $G^*=(V^*,E^*)$, для которого справедливо отношение $G^* \approx G$.

Изоморфизм орграфа на себя называется автоморфизмом и обозначается через ξ. Множество всех автоморфизмов орграфа образует группу по умножению автоморфизмов и обозначается через $Aut(G)$. Порядок группы обозначим через $|Aut(G)|$. Под числом *канонических изоморфных вложений* G^* в G будем понимать величину, определяемую следующим образом $w(G^*,G)=W(G^*,G)/|Aut(G)|$, где $W(G^*,G)$ – число всех изоморфных вложений G^*

в G. Орбитой вершины $v \in V$ называется подмножество $\theta(Aut(G),v)$ вершин орграфа G, которые могут быть отображены на вершину v:

$$\theta(Aut(G),v) = \{v^*:[\exists\, \xi \in Aut(G): \xi(v^*)=v]\}.$$

Орбиты вершин орграфа определяют классы их эквивалентного расположения.

Абстрактный тип t – произвольный орграф, определённый с точностью до изоморфизма. Группу его вершинных автоморфизмов обозначим через $Aut(t)$. Обозначим множество всех канонических изоморфных вложений абстрактного типа t в G через $F^t(G) = \{f_1^t, f_2^t, \ldots f_m^t\}$, а через $|F^t(G)|$ обозначим количество фрагментов типа t в G. Если на множестве вершин типа фрагмента t и G задана нумерация, то фрагмент f^t орграфа G может быть представлен помеченными фрагментами f^{lt}, когда каждой вершине типа фрагмента t сопоставляется номер вершины орграфа G, которой она соответствует при вложении. Число помеченных фрагментов, представляющих один и тот же фрагмент G, равно порядку группы вершинных автоморфизмов абстрактного типа.

Пусть $Aut^t(G)$ является индуцированным представлением группы $Aut(G)$ и определяет симметрию расположения фрагментов типа t в G. Под *t-автоморфизмом G* будем понимать подстановку ξ' на множестве помеченных фрагментов типа t орграфа G (или канонических изоморфных вложений) $F^{lt}(G)$, индуцированную некоторым вершинным автоморфизмом ξ орграфа G. В процессе индуцирования помеченный фрагмент $f_i^{lt} \in F^{lt}(G)$, представленный каноническим изоморфным вложением $(v_1, v_2, \ldots, v_n)$ орграфа G, переходит в помеченный фрагмент $f_j^{lt}(G) \in F^{lt}(G)$, каноническое представление которого получено *канонизацией* вложения $(u_1, u_2, \ldots, u_n)$, где $u_i = \xi(v_i)$, $i = 1..n$, n – число вершин фрагмента типа t. *Группой t-автоморфизмов G* ($Aut^t(G)$ или t-группа) будет группа подстановок, носителем которой является всё множество t-автоморфизмов для данного t, а групповой операцией – операция произведения подстановок. Тот факт, что множество t -автоморфизмов образует группу, непосредственно следует из свойств $Aut(G)$. Степень t-группы равна числу канонических изоморфных вложений абстрактного типа t в орграф G ($|F^{lt}(G)|$), а порядок меньше или равен порядку $Aut(G)$, так как два различных нетождественных вершинных автоморфизма могут индуцировать один и тот же t-автоморфизм. Все понятия, связанные с анализом $Aut^t(G)$, определяются аналогично понятиям, связанным с анализом $Aut(G)$ (например, орбиты t-группы). t-группа точно характеризует симметрию расположения фрагментов типа t в орграфе G. Все неопределяемые ниже понятия можно найти в [12, 13].

3. Задачи различения орграфов и различения расположения их фрагментов

Пусть $\Re_R$ обозначает множество всех связных орграфов, на котором задано отношение R – «быть изоморфными». Тогда *задача различения* (установления факта изоморфизма) двух орграфов G_i, $G_j \in \Re$ определяется следующими параметрами:

$$DG = \langle \Re;\, G_i, G_j;\, R \rangle.$$

Решением задачи DG является ответ на вопрос, являются ли G_i и G_j изоморфными?

Задача *различения расположения двух фрагментов* f_i^{lt}, f_j^{lt} типа t в орграфе G определяется следующими параметрами:

$$DF^t = \langle G;\, f_i^{lt}, f_j^{lt};\, \xi^t \rangle,$$

где ξ^t – отношение принадлежности фрагмента к одной орбите t-группы. Решением задачи является ответ на вопрос, принадлежат ли f_i^{lt}, f_j^{lt} одной и той же орбите t-группы?

4. Инварианты и их применение для решения задач различения орграфов

Основным инструментом при решении задач различения орграфов является понятие инварианта орграфа. Пусть Q_τ – непустое множество с отношением эквивалентности τ (множество чисел, векторов, матриц, графов и т.д.). Функция IN, заданная на множестве $\Re_R(G)$ и принимающая значения в Q_τ, называется *инвариантом орграфа G*, если справедливо условие:

$$\forall G_i, G_j \in \Re[G_i(R)G_j \rightarrow IN(G_i)(\tau)IN(G_j)].$$

Графы G_i, G_j IN-эквивалентные $(G_i(IN)G_j)$, если $IN(G_i)(\tau)IN(G_j)$.

Инвариант IN называется *полным инвариантом орграфа*, если выполняется условие:

$$\forall G_i, G_j \in \Re[IN(G_i)(\tau)IN(G_j) \rightarrow G_i(R)G_j].$$

Под *чувствительностью инварианта орграфа G в множестве* $T \subseteq \Re$ будем понимать отношение вида: $\alpha(IN,R)=|T(IN)|/|T(R)|$, где $|T(IN)|$ обозначает число классов *IN*-эквивалентности орграфов, а $|T(R)|$ – число неизоморфных орграфов в множестве *T*. Чувствительность инварианта определяет точность решения задачи различения орграфов в множестве *T*. *Границей вырождаемости инварианта IN* в множестве $T \subseteq \Re$ называется наименьшее число вершин $G \in T$, при котором инвариант *IN* становится неполным.

5. Матрица достроек фрагментов как инвариант характеризации орграфа и характеризации расположения его фрагментов

Пусть $F^t(G)=\{F^1, F^2, \dots F^t, \dots F^T\}$ ($F^l(G)=\{F^{l1}, F^{l2}, F^{tt}, \dots F^{lT}\}$) обозначает множество всех собственных фрагментов (помеченных фрагментов) орграфа *G*, а $F^{lt}(G)=\{f_1^{lt}, f_2^{lt}, \dots f_j^{lt}, \dots f_{rt}^{lt}\}$ множество помеченных фрагментов типа *t*, *j* – номер фрагмента, *rt* – число фрагментов типа *t*, *T* – число типов фрагментов. Пусть $B=<b_1,b_2,\dots,b_j,\dots,b_{k1}>$ обозначает базис структурных дескрипторов (СД), определяющий точность характеризации орграфа, и пусть $m_{i,j}$ – число достроек фрагмента f_i^{lt} орграфа *G* до фрагмента орграфа, изоморфного $b_j \in B$. Построим матрицу $EM(G/B)=\|m_{i,j}\|$, где $i = 1..|F^l(G)|$, $j=1..k1$. Выберем базис путей, как первоочередной для исследования, что обосновано следующими причинами: (1) любые орграфы включают пути; (2) пути отражают как *локальные* (пути малой длины), так и *глобальные* (пути большой длины) свойства орграфов; (3) пути – наиболее *легко конструктивно перечислимые фрагменты* в сравнении с другими фрагментами орграфов; (4) пути относятся к классу монотонно *наращиваемых по сложности* базисов СД для характеризации орграфов.

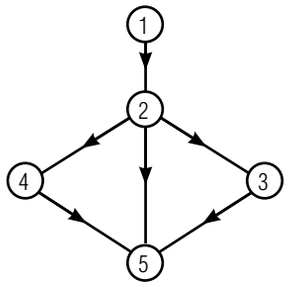


Рис. 1.

Для орграфа *G (рис. 1)* приведены примеры двух видов матриц $EM(G/V \subseteq P)$ *(табл. 1, выделены серым цветом)*, в которых в качестве строк выступают вершины *G*, а в качестве базиса СД – базис путей $P=<P_0, P_1,\dots, P_{(p-1)}>$. Матрица, упорядоченная лексикографически по убыванию значений строк (например, справа налево) является инвариантом, если не сохраняется нумерация вершин *G*. В первом виде матриц достройка выполняется от вершин *как концевых в пути*, а во втором – достраиваемый путь *включает вершину* в любом месте пути. Учитывая, что пути можно достраивать до путей-подграфов (*P*) или путей порожденных подграфов (P^s), получаем четыре вида основных матриц достроек путей до путей:

Таблица 1.

Вершина концевая достраивается до путей						
V	P_0	P_1	P_2	P_3	*irc* ($f^t(C.n)$)	*irc* ($f^t(C)$)
5	1	3	3	2	0,3846	0,3846
1	1	1	3	2	0,3616	0,3616
2	1	4	2	0	0,1231	0,1231
3	1	2	1	0	0,0654	0,1308
4	1	2	1	0	0,0654	
Slw	5	12	10	4	31	1
Sw	1	2	2	2	7	
WF	5	6	5	2	18	
ISC	5	18	45	62	*ISC*(*G/P*)=130	

Вершина достраивается до путей						
V	P_0	P_1	P_2	P_3	*irc* ($f^t(C.n)$)	*irc* ($f^t(C)$)
2	1	4	5	2	0,2885	0,2885
5	1	3	3	2	0,2308	0,2308
1	1	1	3	2	0,2077	0,2077
3	1	2	2	1	0,1365	0,2731
4	1	2	2	1	0,1365	
Slw	5	12	15	8	40	1
Sw	1	2	3	4	10	
WF	5	6	5	2	18	
ISC	5	18	45	62	I*SC*(*G/P*)=130	

$$EM(G/P^l \subseteq^e P);\ EM(G/P^l \subseteq P);$$
$$EM(G/P^l \subseteq^{Se} P);\ EM(G/P^l \subseteq^S P).$$

В наиболее общем виде матрица $EM(G/F^l \subseteq F)$ в качестве строк включает все подграфы из множества $F^l(G)$, а в качестве столбцов – все подграфы из множества $F(G)$. Таким образом, получаем стратифицированную систему матриц от $EM(G/V \subseteq P)$ до $EM(G/F^l \subseteq F)$, расширяемую по трем направлениям: (1) от V до F^l; (2) от P_0 до F; (3) одновременно по (1) и (2). Это приводит к формированию и исследованию *новых систем уточняемых отношений эквивалентности орграфов*.

Чтобы упорядочить фрагменты из $F(G)$ по возрастанию сложности, определим индекс сложности орграфа. Под структурным спектром G в базисе СД $B=<b_1, b_2, ..., b_i,..., b_{k1}>$ будем понимать запись следующего вида $SS(G/B) = \beta_1 b_1, \beta_2 b_2, ..., \beta_i b_i, \beta_{k1} b_{k1})$, где $\beta_i=1$, если фрагмент $b_i \in B$ изоморфно вкладывается в G и $\beta_i=0$ в противном случае [14].

Пусть для G построен его *полный структурный спектр* (ПСС) в базисе СД

$$B=<b_1, b_2, ..., b_i,..., b_{k1}>: FS(G/B)=$$
$$=<w_1 b_1, w_2 b_2, ..., w_i b_i,..., w_{k1} b_{k1}>,$$

где b_i – фрагмент базиса; w_i – число канонических изоморфных вложений фрагмента b_i в G. Очевидно, что $w(K_1)=p$, а $w(K_2)=q$. Примем $ISC(K_1)=1$, $ISC(K_2)=3$. Так как для любого фрагмента f_i орграфа G можно определить его ПСС, а для каждого фрагмента от фрагмента G, можно построить его ПСС и т.д., то рекурсивным методом всегда можно вычислить *индекс спектральной сложности* (ISC) (ниже *сложности*) и вектор-индекс сложности (V_ISC) орграфа G в базисе СД B:

$$ISC(G/B) = w_1 ISC(b_1) +$$
$$+ w_2 ISC(b_2),..., w_j ISC(b_j),..., w_{k1} ISC(b_{k1});$$

$$V_ISC(G/B) =< w_1 ISC(b_1);\ w_2 ISC(b_2);\ ...;$$
$$w_j ISC(b_j);\ ...;\ w_{k1} ISC(b_{k1}) >.$$

Для G (*рис. 1*) при использовании базиса путей-подграфов $P=<P_0, P_1, P_2, P_3>$ получим:

- ✦ $ISC(G/P)=5 \cdot 1+6 \cdot 3+5 \cdot 9+2 \cdot 31=130$;
- ✦ $V_ISC(G/P)=<5; 18; 45; 62>$;
- ✦ $SS(G/P)=<1, 1, 1, 1>$;
- ✦ $FS(G/P)=<5, 6, 5, 2>$.

6. Расширенная матрица достроек фрагментов и матрица вкладов фрагментов в общую сложность орграфа

Эта матрица определяет основу иерархического метода анализа сложности орграфа с вычислением вкладов фрагментов, вкладов каждого класса эквивалентно расположенных фрагментов для каждого типа t и вкладов фрагментов в сложность орграфа.

Добавим к матрице $EM(G/F^l \subseteq B)$ пять новых строк:

1. Строки с номером нуль для вектор-индекса сложности элементов $b_j \in B$:

$$V_ISC(G/B) =< ISC(b_1),..., ISC(b_j),..., ISC(b_{k1}) >.$$

2. Строки с номером (k+1) со значениями элементов

$$Slw(F^l/b_j) = \sum_{f^{lt} \in F^l} \sum_{i=1}^{rt} m_{ij}.$$

3. Строки с номером (k+2), значения элементов которой $Sw(b_j)$, равны суммарному числу канонических изоморфных вложений фрагмента типа t в $b_j \in B$ по всем типам $t = 1,2, ..., T$, т.е.

$$Sw(F^l/B) =< Sw(F^l/b_1),..., Sw(F^l/b_j),..., Sw(F^l/b_{k1}) >,$$

где $$Sw(F^l/b_j) = \sum_{t=1}^{T} \frac{f^t(b_j)}{\left|Aut(f^t)\right|} = \sum_{t=1}^{T} w(f^t/b_j).$$

4. Строки с номером (k+3) со значениями элементов

$$w_j(b_j) = Slw(F^l/b_j)/Sw(F^l/b_j).$$

5. Строки с номером (k+4) со значениями элементов

$$V_ISC(G/b_j) = w_j(b_j) ISC(b_j).$$

В результате построим расширенную матрицу достроек фрагментов

$$EM^*(G/F^l \subseteq B).$$

На основе $EM^*(G/F^l \subseteq B)$ построим *матрицу*

$$MIRC(F^l \subseteq B(G)) = \left\| irc(f_i^t/b_j) \right\|,\ i = 0..k+4;\ j = 1..k1+3$$

относительных вкладов фрагментов в сложность G.

Значения элементов матрицы вычисляются по формуле

$$irc(f_i^t / b_j) = \frac{m_{ij}}{Sw(F^l / b_j)} \frac{ISC(b_j)}{ISC(G/B)}.$$

Тогда величина $irc(f_i^t / B)$, вычисляемая по формуле

$$irc(f_i^t / B) = \frac{1}{ISC(G/B)} \sum_{j=1}^{k1} m_{ij} \frac{ISC(b_j)}{Sw(F^l / b_j)},$$

определяет относительный вклад f_i^t в общую сложность при использовании базиса B.

Фрагменты f_i^t, имеющие одинаковые значения вкладов $irc(f_i^t / B)$, образуют класс $f^t(C)$ эквивалентных по расположению фрагментов типа t с общим вкладом $irc(f_i^t(C)/B)$. Сумма относительных вкладов по всем фрагментам одного типа t образует вклад $irc(f^t / B)$. Расширенную матрицу достроек фрагментов, дополненную тремя столбцами:

✦ (k1+1) со значениями $irc(f^t(C.n)/B)$ для каждого типа t;

✦ (k1+2) со значениями $irc(f^t(C)/B)$ для каждого типа t;

✦ (k1+3) со значениями $irc(f^t / B)$ для каждого типа t,

обозначим $MIRC(F^l \subseteq B(G))$.

Пример матрицы $MIRC(V \subseteq P(G))$ без значений $irc(f^t(C.n)/b_j)$ для орграфа G (*рис. 1*) приведен в *табл. 1*. На основе $MIRC(F^l \subseteq F(G))$ легко построить *матрицу абсолютных вкладов фрагментов в сложность G*, т.е. матрицу $MIAC(F^l \subseteq F(G))$.

7. Метод иерархического анализа сложности орграфов

Вычисление матриц $MIRC(F^l \subseteq B(G))$ позволяет проводить *иерархический анализ* сложности, включающий пять основных уровней, с последовательным уточнением результатов на основе:

1. Индексов сложности – $ISC(G/B)$;

2. Вектор-индексов сложности – $V_ISC(G/B)$;

3. Вектор-индексов относительных вкладов *каждого типа фрагментов t* (вершины, дуги, пути с длиной 2 и т.д.) – $V_irc(G, F^t)$, где $t = 1..T$;

4. Вектор-индексов относительных вкладов *классов фрагментов в каждом типе* фрагментов

$$V_irc(G, F^t(C)),$$

где C – номер класса фрагментов типа t и $t = 1..T$;

5. Вектор-индексов относительных вкладов каждого *фрагмента с номером n в классе C типа t* –

$$V_irc(F^t(C.n)/B), \text{ где } t = 1..T.$$

Дополнительные уровни анализа сложности определяются на основе *вектор-индексов относительных вкладов* каждого *фрагмента с номером n в классе C типа t* по каждому элементу $b_j \in B$ –

$$V_irc(F^t(C.n)/b_j), \text{ где } t=1..T, j=k1, (k1-1),..., 1.$$

Для трех орграфов (*рис. 2*) в *табл. 2* приведены матрицы $MIAC(P^l \subseteq P(G))$, упорядоченные по убыванию значений вкладов всех путей по каждому типу t, где $t=P_0 - P_2$.

Приведем результаты сравнительного анализа трех орграфов по иерархической схеме:

✦ на первом уровне

$ISC(G_1/P)=ISC(G_2/P)=ISC(G_3/P)$;

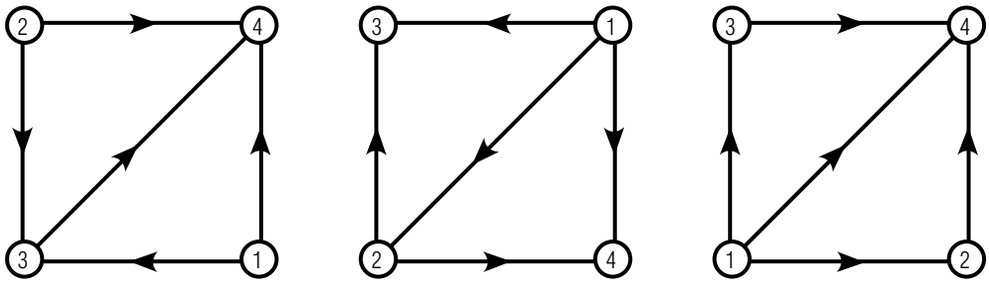


Рис. 2.

Таблица 2.

$MIAC(P^l \subseteq P(G_1)) = MIAC(P^l \subseteq P(G_2))$								$MIAC(P^l \subseteq P(G_3))$						
$P^l(G_1)$	$P^l(G_2)$	1	3	9	irc $(P^t(C.n))$	irc $(P^t(C))$	irc (P^t)	$P^l(G_3)$	1	3	9	irc $(P^t(C.n))$	irc $(P^t(C))$	irc (P^t)
3	2	1	3	3	7	14	23	4	1	3	3	7	14	23
4	1	1	3	3	7			1	1	3	3	7		
1	4	1	2	1,5	4,5	9		3	1	2	1,5	4,5	9	
2	3	1	2	1,5	4,5			2	1	2	1,5	4,5		
34	12	0	1	3	4	4	11	13	0	1	1,5	2,5	10	11
13	23	0	1	1,5	2,5	5		34	0	1	1,5	2,5		
23	24	0	1	1,5	2,5			12	0	1	1,5	2,5		
14	13	0	1	0	1	2		24	0	1	1,5	2,5		
24	14	0	1	0	1			14	0	1	0	1	1	
134	124	0	0	1,5	1,5	3	3	124	0	0	1,5	1,5	3	3
234	123	0	0	1,5	1,5			134	0	0	1,5	1,5		
$Slw(b_j)$		4	15	18	37	37	37	$Slw(b_j)$	4	15	18	37	37	37
$Sw(b_j)$		1	3	9	10	$ISC(G/P)$=37		$Sw(b_j)$	1	3	9	10	$ISC(G/P)$=37	
$WF(b_j)$		4	5	2	11			$WF(b_j)$	4	5	2	11		
$V_ISC(b_j)$		4	15	18	37			$V_ISC(b_j)$	4	15	18	37		

✦ на втором уровне

$V_ISC(G_1/P)=V_ISC(G_2/P)=V_ISC(G_3/P)$;

✦ на третьем уровне

$V_irc(G_1,P^t(C))=V_irc(G_2,P^t(C))=V_irc(G_3,P^t(C))$;

✦ на четвертом уровне

$V_irc(G_1,P^t(C.n)/B)=$
$=V_irc(G_1,P^t(C.n)/B)\neq V_irc(G_1,P^t(C.n)/B)$.

Таким образом, на четвертом уровне выявлено различие по сложности орграфов G_1 и G_2 с G_3. Для G_1 и G_2 значения матриц $MIAC(P^l \subseteq P(G))$ одинаковые, следовательно, G_1 и G_2 IN-эквивалентные орграфы по $MIAC(P^l \subseteq P(G))$. Заметим, что полученные классы эквивалентно расположенных путей в G_1–G_3, являются орбитами групп

$$Aut^{P_0}(G),\ Aut^{P_1}(G),\ Aut^{P_2}(G).$$

Если базис путей является достаточным для того, чтобы классы эквивалентно расположенных путей стали орбитами групп

$$Aut^{P_0}(G),\ Aut^{P_1}(G),\ldots,\ Aut^{P_{(p-1)}}(G),$$

то нетрудно доказать, что справедливы следующие утверждения.

Утверждение 1. Индекс $TIIC(G/V)$ и индексы $TIIC(G/P_1),\ldots,TIIC(G/P_{(p-1)})$ являются восстанавливаемыми характеристиками из $MIRC(P^l \subseteq P(G))$ для орграфа G.

Действительно, например, для орграфа G_1 (*рис. 2*) имеем

$$TIIC(G_1/V)=WF(b_1)\left(\frac{|P^0(C_1)|}{WF(b_1)}\log_2(|P^0(C_1)|)+\right.$$
$$\left.+\frac{|P^0(C_2)|}{WF(b_1)}\log_2(|P^0(C_2)|)\right).$$

Утверждение 2. Индекс $OIIC(G/P_2)$ и индексы $OIIC(G/P_3),\ldots,OIIC(G/P_{(p-1)})$ являются восстанавливаемыми характеристиками из $MIAC(P^l \subseteq P(G))$ для орграфа G.

Действительно, например, для орграфа G_1 (*рис. 2*) имеем

$$OII(G/P_2) = 2WF(b_2)\Big(|P^2(C_1)|\log_2(|P^2(C_1)|)+$$
$$+|P^2(C_2)|\log_2(|P^2(C_2)|)+|P^2(C_3)|\log_2(|P^2(C_3)|)\Big).$$

Утверждение 3. Индекс $CM(G)$ являются восстанавливаемой характеристикой из $MIRC(P^l \subseteq P(G))$ для орграфа G.

Действительно, например, для орграфа G_1 (*рис. 2*) имеем

$$OII(G_1/P_2) = \frac{WF(b_1)WF(b_2)}{WF(b_1)+WF(b_2)}\big(WF(b_2)+WF(b_3)\big).$$

Если классы расположения путей не являются орбитами, то мы имеем метод *приближенного вычисления* рассмотренных индексов. Для точного вычисления индексов необходимо базис путей наращивать элементами типа полупути, контура, ордеревья и т.д. Выделим, что добавляя к базису путей, например, один полупуть вида $K_{1,2}$, мы построим $MIRC(P^l \subseteq (P \cup K_{1,2})(G))$ и определим не изоморфизм G_1 и G_2 (*рис. 2*).

8. Результаты определения точности решения задач различения орграфов

На *рис. 3* приведены диаграммы всех орграфов с p=3.

Сравнивая орграфы по значениям относительных вкладов путей в сложность ($irc(P^l(C.n))$) из матриц $MIRC(P^l \subseteq P(G))$, получим: 10 классов:

$$(G_1)>(G_2)>(G_3)>(G_4)>(G_5)>(G_6,G_7)>(G_8,G_9)>(G_{10})>(G_{11})>(G_{12},G_{13}).$$

Следовательно, точность решения задачи различения орграфов

$$\alpha(MIRC(P^l \subseteq P(G)), R) = 0,769.$$

Заметим, что все пары орграфов, образующие классы эквивалентности, имеют инверсную ориентацию дуг. Такие орграфы будем называть *инверсными*. Для них справедливо

Утверждение 4. Неизоморфные инверсные пары орграфов являются *IN*-эквивалентными по значениям матрицы $MIRC(P^l \subseteq P(G))$.

Следовательно, справедливо

Утверждение 5. Матрица $MIRC(P^l \subseteq P(G))$ не является полным инвариантом орграфов и граница ее вырождаемости равна трем.

В *табл. 3* приведены результаты определения точности решения задач различения для орграфов, сетей, графов по $ISC(G/P)$ и $MIRC(P^l \subseteq P(G))$.

Использованы следующие обозначения: *ND*, *NS*, *NG* – число соответственно орграфов, сетей, графов; $\alpha(D,ISC)$, $\alpha(S,ISC)$, $\alpha(G,ISC)$ – чувствительность по *ISC* соответственно для орграфов, сетей, графов; $\alpha(D,MIRC)$, $\alpha(S,MIRC)$, $\alpha(G,MIRC)$ – чувствительность по *MIRC* соответственно для орграфов, сетей, графов.

Анализ результатов приводит к следующим выводам: (1) граница вырождаемости $ISC(G/P)$ для

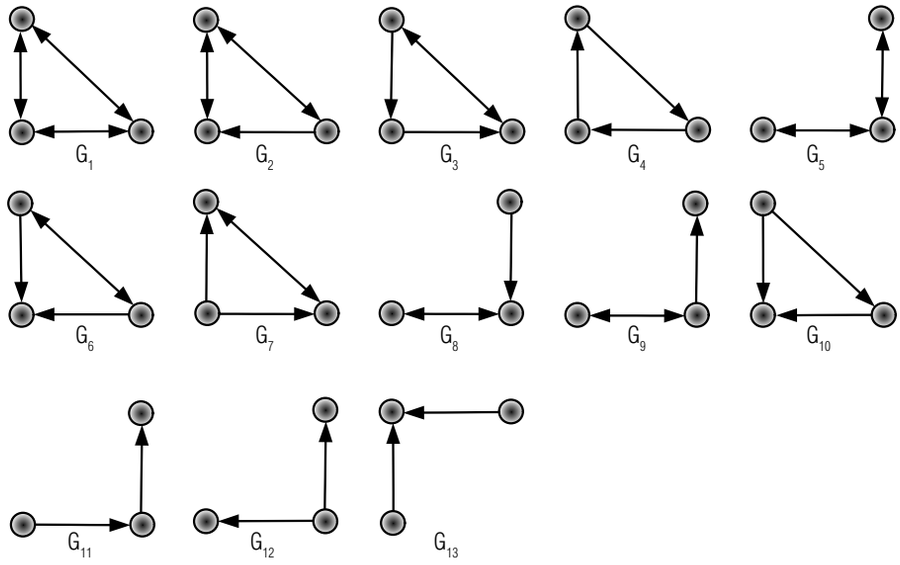


Рис. 3.

Таблица 3.

p	*ND*	α (*D*,*ISC*)	α (*D*,*MIRC*)	*NS*	α (*S*,*ISC*)	α (*S*,*MIRC*)	*NG*	α (*G*,*ISC*)	α (*G*,*MIRC*)
3	13	0,615	0,769	4	0,750	0,75	2	1	1
4	199	0,407	0,638	24	0,542	0,64	6	1	1
5	9364	0,183	0,533	267	0,269	0,51	21	0,947	1
6	1530843	0,064	0,596	5647	0,094	0,47	112	0,938	1
7				237317	0,019	0,47	852	0,977	1
8							11117	0,983	1
9							261080	0,989	1

орграфов (сетей, графов) равна 3 (3, 5); (2) граница вырождаемости $MIRC(P^l \subseteq P(G))$ для орграфов (сетей, графов) равна 3 (3, больше 9); (3) каждый класс эквивалентности орграфов включает инверсные пары орграфов; (4) для более точного решения задач различения орграфов (сетей) необходимо базис путей расширять полупутями.

9. Новые методы анализа сходства орграфов

Задача вычисления сходства орграфов включает следующие параметры:

1. $M=\{G_1, G_2,..., G_i,..., G_n\}$ – множество орграфов, анализируемых на сходство.

2. $B=<b_1, b_2,..., b_j,..., b_k>$ – базис СД или правила его определения.

5. $ISC(B)=<ISC(b_1); ISC(b_2); ...; ISC(b_j); ISC(b_{k1})>$ – набор весов СД для учета их качественных характеристик.

4. $D(G_i,G_j)$ – метрика (псевдометрика) или функция для вычисления коэффициентов сходства (несходства) (*SC*).

Необходимо построить матрицу (граф) попарных расстояний или коэффициентов сходства (несходства).

Выделим два метода решения задачи на основе матричных моделей:

1. Метод иерархического уточняющего анализа, использующий в качестве метрик:

✦ на первом уровне – $|ISC(G_i / B) - ISC(G_j / B)|$;

✦ на 2-5 уровнях – метрику Евклида для соответствующих вектор-индексов вкладов из $MIRC(F^l \subseteq B(G))$, для каждого класса, полученного на предыдущем уровне.

2. Метод анализа на основе подструктурного подхода (ПП), использующего вычисление попарных расстояний

$$D(G_i,G_j) = \left|V(F^l \subseteq B(G_i))\right| + \left|E(F^l \subseteq B(G_i))\right| + \left|V(F^l \subseteq B(G_j))\right| + \left|E(F^l \subseteq B(G_j))\right| - 2\left|V(mcf(F^l \subseteq B(G), F^l \subseteq B(G_j)))\right|,$$

где $F^l \subseteq B(G)$ – двудольная граф-модель с весами на дугах, соответствующая $EM(G/F^l \subseteq F)$, $mcf(F^l \subseteq B(G_i), F^l \subseteq B(G_j))$ – *максимальный общий подграф* по числу дуг для двух граф-моделей $F^l \subseteq B(G_i)$, $F^l \subseteq B(G_j)$ или коэффициент сходства

$$MSI(G_i,G_j) = (\left|V(mcf(F^l \subseteq B(G_i), F^l \subseteq B(G_j)))\right| + \left|E(mcf(F^l \subseteq B(G_i), F^l \subseteq B(G_j)))\right|^2 / (\left|V(F^l \subseteq B(G_i))\right| + \left|E(F^l \subseteq B(G_i))\right|) \cdot (\left|V(F^l \subseteq B(G_j))\right| + \left|E(F^l \subseteq B(G_j))\right|).$$

Приведем результаты анализа сходства на основе $EM(G/P^l \subseteq {}^S F)$ по второму подходу для 13 орграфов (*рис. 3*) и результаты их кластеризации. В графе попарных расстояний получено 19 разных значений расстояний (весов ребер) в диапазоне от 0 до 48. При кластеризации из графа попарных расстояний последовательно удалялись ребра, значения веса которых, превышали соответственно следующие границы 9; 8; 6; 3; 2; 0. В результате выделено 7 одноэлементных кластеров и 3 двухэлементных, состоящих из *неизоморфных инверсных орграфов* (*рис. 4*). Такие же кластеры выделены на этапе 4 по методу подхода 1.

Заметим, что использование ПП, основанного на определении *mcf* для всех пар орграфов *при одинаковом числе вершин*, не позволяет проводить кластеризацию, из-за недостаточного диапазона значений весов ребер в графах попарного сходства.

Выделим, что, используя метрику Евклида для вычисления попарных расстояний между строками матрицы $MIRC(F^I \subseteq F(G))$, мы решаем задачу определения попарного сходства *расположения фрагментов орграфа*.

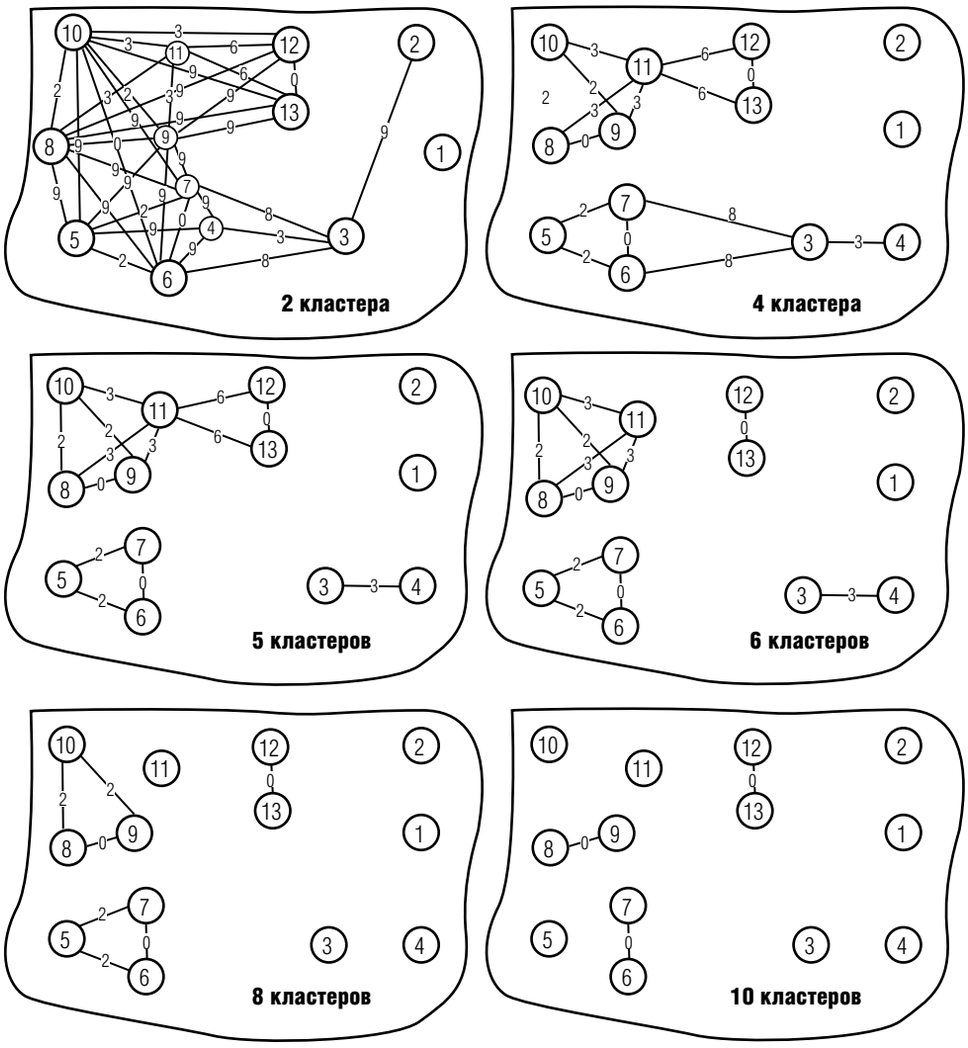


Рис. 4.

10. Экспериментальные оценки вычислительной сложности задачи построения матрицы относительных вкладов фрагментов в сложность орграфов

Экспериментальные оценки вычислительной сложности (ЭОВС) решения задачи построения матрицы $MIRC(P^I \subseteq P(G))$ получены на основе использования новых программных подсистем, включенных в АСНИ «Мастерская граф-моделей» [16, 17]. Эксперименты проводились на ноутбуке (SONY, Intel Core i5, 2.67 GHz, память 4 *Gb*, Windows 7).

Учитывая особенность ордеревьев, связанную с существованием для каждой пары вершин только одного пути, нетрудно получить теоретическую полиномиальную оценку $T=O(p^3)$ вычислительной сложности алгоритма построения $MIRC(P^I \subseteq P(G))$ для ордеревьев. Основу алгоритма составляет метод достройки вершины ордерева до пути, заданной длины. На *рис. 5* приведены графики ЭОВС алгоритма построения $MIRC(P^I \subseteq P(G))$для ордеревьев со средними значениями индексов сложности ($ISC(G/P)$) и числами вершин от 108 до 508.

В общем случае задача построения $MIRC(P^I \subseteq P(G))$ для орграфов (сетей) является *NP*-полной проблемой, так как число путей всех длин в орграфах (сетях) растет экспоненциально и частным случаем для нее является класическая *NP*-полная проблема определения гамильтонова пути в орграфе (сети)

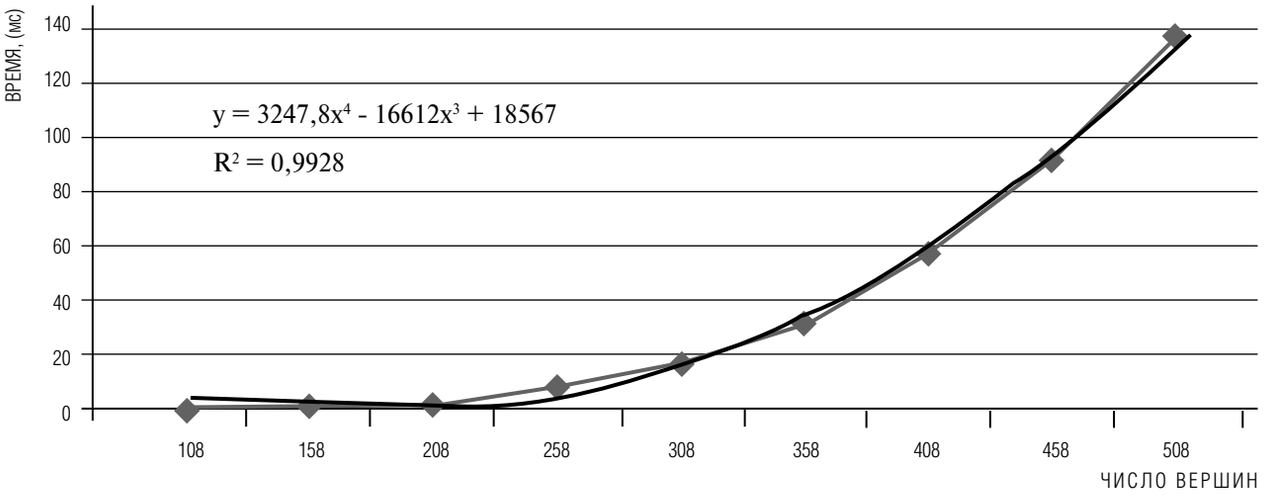


Рис. 5.

[15]. Сокращение перебора при построении матрицы $MIRC(P^l \subseteq P(G))$ методом достройки вершин до путей, заданной длины, осуществляется за счет учета симметрии (орбит стабилизаторов $Aut(G)$) и инвариантов вершин орграфов [13].

На *рис. 6-9* приведены ЭОВС алгоритма построения восьми видов матриц:

$MIRC(P^l_{0\text{-}0} \subseteq {}^sP_{0\text{-}12}(G))$; $MIRC(P^l_{0\text{-}2} \subseteq {}^sP_{0\text{-}12}(G))$;
$MIRC(P^l_{0\text{-}4} \subseteq {}^sP_{0\text{-}12}(G))$; $MIRC(P^l_{0\text{-}6} \subseteq {}^sP_{0\text{-}12}(G))$;

$MIRC(P^l_{0\text{-}0} \subseteq {}^sP_{0\text{-}20}(G))$; $MIRC(P^l_{0\text{-}2} \subseteq {}^sP_{0\text{-}20}(G))$;
$MIRC(P^l_{0\text{-}4} \subseteq {}^sP_{0\text{-}20}(G))$; $MIRC(P^l_{0\text{-}6} \subseteq {}^sP_{0\text{-}20}(G))$;

Анализировались случайные орграфы с числами вершин от 100 до 600 и средней степенью (суммой полустепеней исхода и захода) равной четырем. Были построены матрицы *MIRC* следующих размеров (число строк × число столбцов): 1) *y*1 от 100×13 до 520×13; 2) *y*2 от 622×13 до 3395×13; 3) *y*3 от 1962×13 до 12369×13; 4) *y*4 от 4945×13 до 39675×13; 5) *x*1 от 100×21 до 520×21; 6) *x*2 от 622×21 до 3395×21; 7) *x*3 от 1962×21 до 12369×21; 8) *x*4 от 4945×21 до 39675×21.

Необходимо выделить, что задача вычисления сходства пары орграфов на основе поиска МОФ их матриц $MIRC(P^l \subseteq P(G))$ имеет полиномиальную вычислительную сложность при условии, что используется один и тот же базис *P*. Это обосновано возможностью сведения задачи поиска МОФ двух матриц $MIRC(P^l \subseteq P(G))$ к задаче поиска максимального паросочетания в двудольном орграфе с весами на дугах.

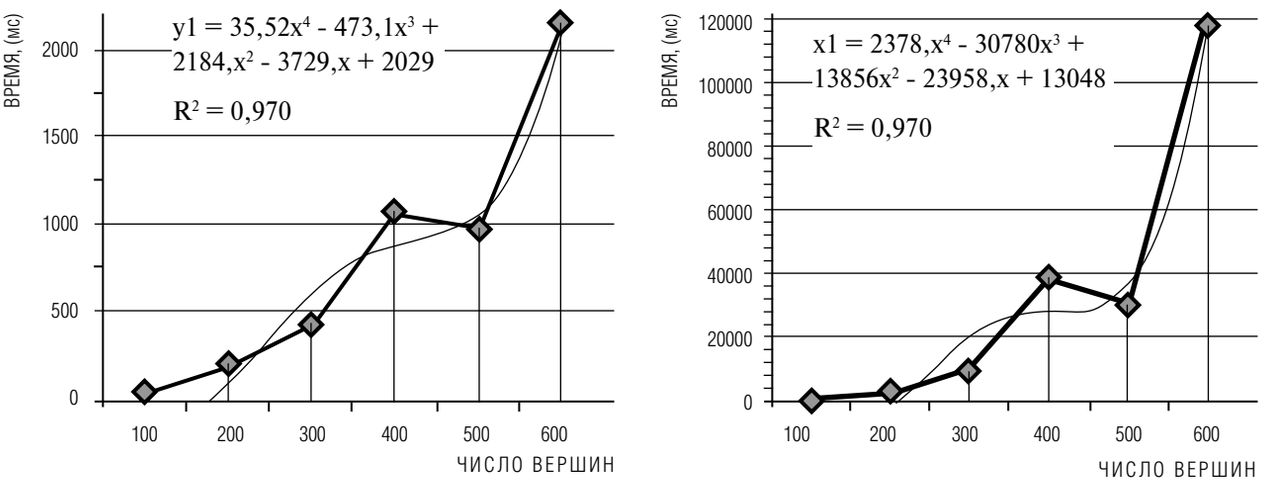


Рис. 6.

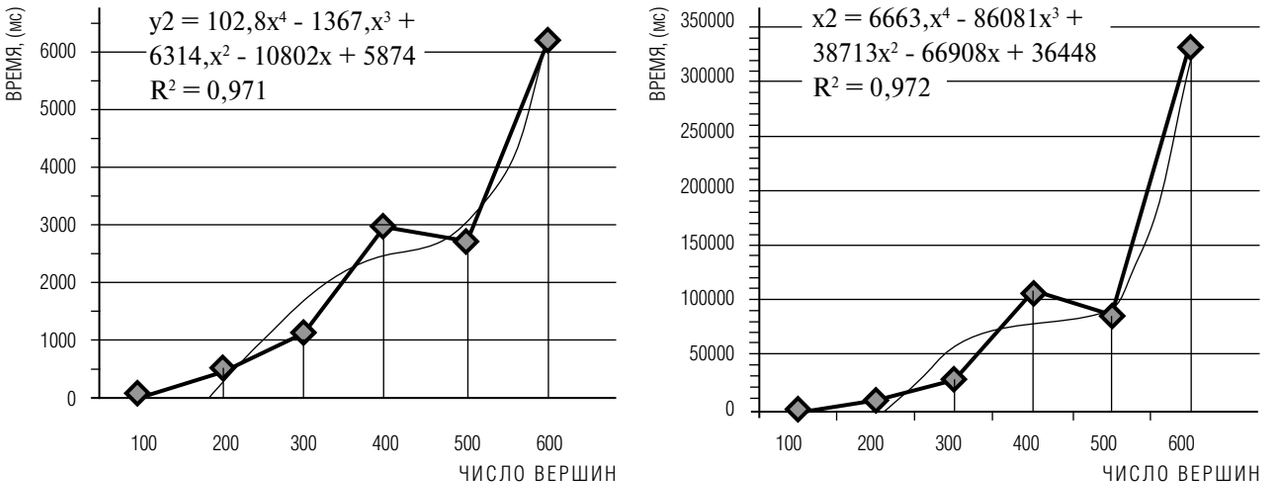


Рис. 7.

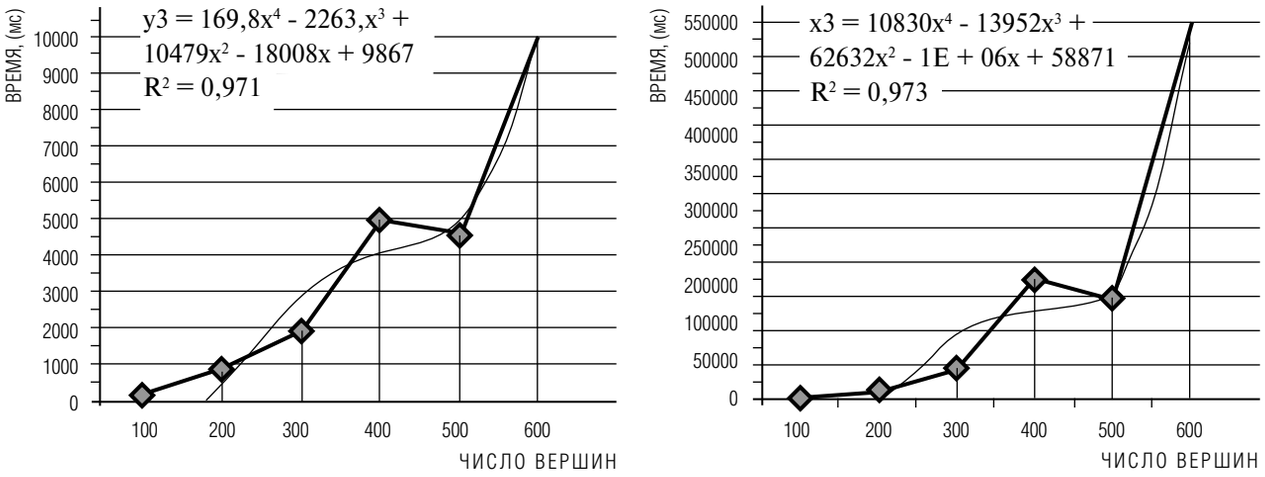


Рис. 8.

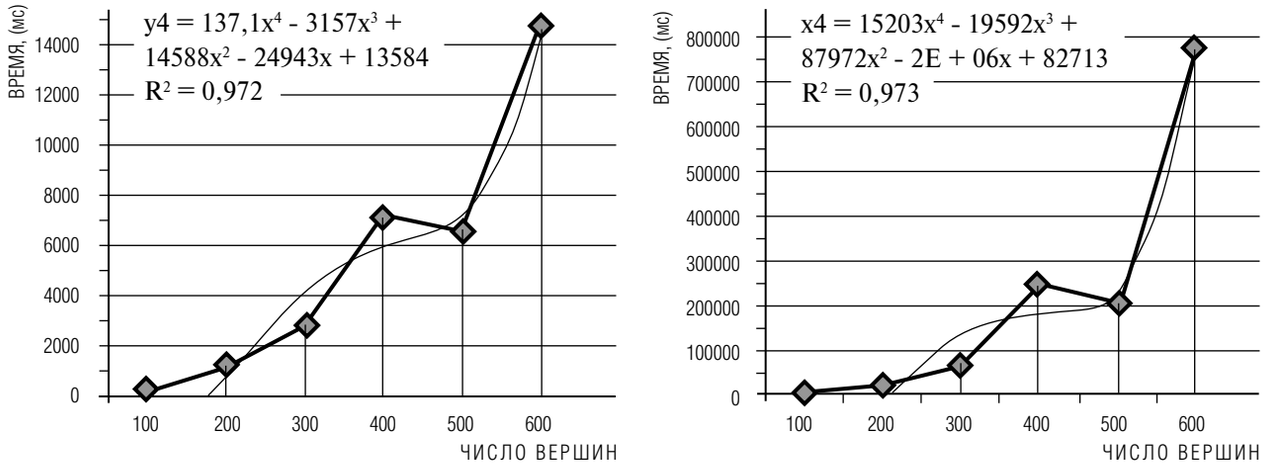


Рис. 9.

Заключение

Предложены матричные модели, позволяющие проводить иерархический уточняющий анализ сложности и сходства с учетом количественных и качественных характеристик фрагментов орграфов. Эти модели позволили расширить и дополнить теоретико-информационный подход к определению сложности орграфов качественными характеристиками его фрагментов. Рассмотрено применение моделей для решения теоретических задач, связанных с определением границ вырождаемости матричных моделей в базисе путей для орграфов, сетей и графов. Модели полезны для решения прикладных задач различения и определения сходства ГМС, например, логико-вычислительных сетей, представляющих знания, сетей коммуникаций, компьютерных сетей и др. Предложенные методы анализа сложности и сходства программно реализованы и используются в учебном процессе НИУ ВШЭ, МЭИ (ТУ). ■

Литература

1. Финн В.К. Индуктивный метод соединенного сходства-различия и процедурная семантика ДСМ-метода.//НТИ, сер. 2, №4, 2010.
2. Кузнецов С.О. Финн В.К. Распространение процедур ЭС типа ДСМ на графы. //Техн.кибернетика, 1988, №5, С.4-11.
3. Игнатов Д.И., Кузнецов С.О. О поиске сходства интернет-документов с помощью частых замкнутых множеств признаков. Десятая национальная конференция по искусственному интеллекту с международным участием. КИИ-2006: Труды конференции. В 3-х т. М.: Физматлит, 2006. С. 245-249.
4. Шеннон К. Работы по теории информации и кибернетике. — М.: Изд. ин. лит., 1963.
5. Бончев Д.Г. Характеризация химических структур с помощью теории информации и теории графов. Автореферат дисс. докт. хим. наук. – Бургас, 1983. – 48 с.
6. Бертц С. Математическая модель молекулярной сложности. Химические приложения топологии и теории графов /Под ред. Р. Кинга, М.: Мир, 1987. – С. 236-258.
7. Minoly D. Combinatorial graph complexity. // Atti. Acad. Waz. Li. Rend. A. Sci. fis. mat. l'natur, vol. 59, no. 6. pp. 154-171, 1975.
8. Grone R., Herris R. A Bound for the Complexity of a Simple Graph. //Discrete Math., vol. 69, no. 1, pp. 97-99, 1988.
9. Загорянская А.А., Кохов В.А. Структурная сложность информационных систем и ее характеризация в базисе деревьев. Труды НТК студентов и аспирантов вузов России, Т1, Москва, 1998. – С.180-182.
10. Bonchev D., Polansky O.E. On the Topological Complexity on Chemical Systems // Studies in Physical and Theoretical Chemistry, 1987.V.51. P. 126-158.
11. Кохов В.А. Концептуальные и математические модели сложности графов. М.: Издательство МЭИ, 2002. – 160 с.
12. Касьянов В.Н., Евстигнеев В.А. Графы в программировании: обработка, визуализация и применение. – Спб.: БХВ-Петербург, 2003. – 1104 с.
13. Алгоритмы и программы решения задач на графах и сетях //Нечепуренко М.И., Попков В.К., Кохов В.А. и др. Новосибирск: Наука. 1990. – 515 с.
14. Кохов В.А. Метод количественного определения сходства графов на основе структурных спектров. Известия АН СССР, сер. Техническая кибернетика. N5, 1994. – С.143-160.
15. Гэри М, Джонсон Д. Вычислительные машины и труднорешаемые задачи. М.: Мир. 1982. – 416 с.
16. Кохов В.А., Ткаченко С.В. Программная система для исследования вычислительной сложности решения задач на графовых моделях. // Программные продукты и системы. N4, 2009. С. 137-140.
17. Незнанов А.А., Кохов В.А. Программный комплекс анализа структурного сходства систем с учетом расположения фрагментов. // Программные продукты и системы. N4, 2009. С. 147-150.

КОМПОНЕНТНАЯ АРХИТЕКТУРА С ОПРЕДЕЛЕНИЕМ ТИПОВ ВО ВРЕМЯ ИСПОЛНЕНИЯ

Е.М. Гринкруг,
кандидат технических наук, доцент кафедры Управления разработкой программного обеспечения Национального исследовательского университета «Высшая школа экономики»,
e-mail: egrinkrug@hse.ru.

А.Р. Шакуров,
аспирант кафедры Управления разработкой программного обеспечения Национального исследовательского университета «Высшая школа экономики»,
e-mail: amir-shak@yandex.ru.
Адрес: г. Москва, ул. Кирпичная, д. 33/5.

В работе рассматриваются вопросы компонентного подхода к разработке программного обеспечения. На основе анализа основных идей данного подхода, их реализации в существующих на сегодняшний день технологиях, присущих им ограничений и перспективных направлений их развития предлагается новая компонентная архитектура, расширяющая функциональные возможности существующих компонентных моделей. Описываются основные принципы построения данной архитектуры.

Ключевые слова: компонент, компонентная разработка, интерфейс, реализация.

1. Введение

Разработка программного обеспечения — трудоёмкий процесс, и неудивительно, что *повторному использованию кода* [8] придаётся большое значение. Многократное использование реализованной функциональности снижает трудозатраты при разработке больших программных систем, сокращает количество ошибок в коде и упрощает его сопровождение.

Наиболее распространёнными примерами повторного использования кода являются программные библиотеки, паттерны проектирования [5] и каркасы приложений. Объектно-ориентированное и обобщённое программирование [4] также содержат в своей основе данную идею. Однако наиболее многообещающей её формой нам представляется *компонентный подход* к созданию программного обеспечения [9].

Под программным компонентом в общем случае понимают некоторую сущность, содержащую данные и реализующую функциональность, скрытые за чётко определёнными интерфейсами (ср. [10] и [11]). Взаимодействие с подобным «чёрным ящиком» осуществляется исключительно через интерфейс. Более точное определение компонента приведено в разделе 4.

Само понятие «интерфейса» трактуется поразному в различных технологиях. Это может быть совокупность «свойств» («атрибутов», «членов» и т.п.) [13] или самостоятельная неделимая сущность [2]. Возможны и другие варианты. Ниже мы предложим оптимальный, на наш взгляд, подход и приведём ряд аргументов в его пользу.

В рамках компонентной идеологии соединение компонентов в работоспособную систему полагается сравнительно простой операцией (см., напр., [3]). Но манипулирование структурой компонентной системы осуществляется в разных технологиях различными способами. Наша задача — найти оптимальный вариант организации взаимодействия компонентов, представляющий собой сбалансированное между гибкостью и простотой использования решение.

2. Ограничения существующих компонентных моделей

Несмотря на очевидные преимущества компонентного подхода, его реализации в тех или иных технологиях на сегодняшний день имеют ряд существенных ограничений.

Основная сложность, по-видимому, заключается в том, что сторонний код всегда либо недостаточно универсален, либо содержит избыточную функциональность. Последнее характерно для программных библиотек, т.к. универсальность — критически важное для них свойство. Но избыточность приводит к неоправданному «раздуванию» итоговой системы, усложнению процесса разработки и т.п.

Компонент может быть сделан более специализированным, т.к. он структурно меньше, чем целая библиотека. Но и компромисс между универсальностью и специализированностью здесь более важен: если иногда есть возможность использовать лишь часть библиотеки, то часть компонента использовать не удастся, не нарушив инкапсуляции.

Одним из способов достижения указанного компромисса является разделение работы с компонентом на проектирование и разработку (designtime) и использование конечным приложением (runtime). Необходимость такого разделения признана и до некоторой степени реализована (см., напр., [13]), однако, на наш взгляд, в этой области возможно достичь большего.

Приведём пример. Пусть, проектируя приложение с графическим интерфейсом, мы должны изменить надпись на компоненте «кнопка». Изменение окончательно: после запуска программы надпись меняться не будет. Реализации данного процесса (запись в соответствующее свойство необходимой строки) в различных технологиях обладают общим недостатком: переменное свойство остаётся изменяемым и после запуска приложения. Т.о., хотя заранее известно, что надпись на кнопке во время исполнения должна быть константой, реализовать это существующими средствами невозможно.

На данную проблему можно взглянуть иначе. Сложность глубокой подстройки компонента под контекст использования (разделение «designtime — runtime» — лишь частный пример) связана с тем, что фактически речь идёт об определении нового типа данных. И т.к. любые манипуляции с компонентом предполагают использование его функциональности и, следовательно, его запуска, определение нового типа должно осуществляться во время исполнения (и без перекомпиляции).

Существуют обходные пути: генерация кода (так поступает большая часть инструментов визуального проектирования GUI (напр., [1])) и вызов компилятора, непосредственная генерация байт-кода и т.п. Однако компилятор доступен далеко не во всякой среде исполнения; данное ограничение, в частности, свойственно встраиваемым системам. По этой причине представляются интересными системы, не прибегающие к помощи компиляторов и т.п. обходных путей и способные создавать из компонентов, настроенных по усмотрению пользователя, новые типы, оставаясь при этом в рамках собственной компонентной модели.

3. Структуризация данных и управление потоком исполнения

Рассмотрим основные аспекты распространённых сегодня языков программирования и компонентных технологий с целью выявить общие идеи, лежащие в их основе, и найти наиболее подходящие реализации каждой из них. Анализ определит основные черты предлагаемой модели, описание которой приведено в следующем разделе.

Детальное рассмотрение конкретных языков и технологий и их отличительных особенностей выходит за рамки данной работы (подобный обзор можно найти в [12]). Мы отталкиваемся от анализа объектно-ориентированной концепции в целом, приводя частные примеры и привлекая технологии, выходящие за рамки объектно-ориентированной парадигмы, когда это необходимо, т.к. требование возможности определения новых типов во время исполнения приводит к необходимости построения компонентной технологии, где время разработки и время исполнения не имеют строгого разграничения.

Всякая среда разработки/исполнения может быть рассмотрена с точки зрения предлагаемых ею принципов и механизмов 1) организации данных и 2) управления потоком исполнения. С точки зрения первого аспекта для объектно-ориентированной программы отметим возможность иерархической организации данных. Это проверенное средство борьбы с нарастающей сложностью программных систем, поэтому предлагаемая компонентная модель предполагает возможность объединения компонентов во взаимодействующие группы, представляющие собой новые компоненты, и т.д. (строго говоря, в группы объединяются *типы компонентов,* после чего создаются сами составные компоненты, см. раздел 4).

В части управления потоком исполнения, как нам кажется, объектно-ориентированный подход не столь удачен. Взаимодействие объектов осуществляется посредством методов, но концепция метода чересчур гибка, что приводит к усложнению структуры программ, использующих её. Метод может возвращать значение, а может не возвращать его. Возвращаемое значение может быть ссылкой на внутреннее поле объекта, но может являться и его защитной копией. Он может иметь произвольное количество параметров произвольных типов, а может не иметь их вовсе. Список можно продолжить. Понятие метода не предоставляет возможности достаточной формализации взаимодействия объектов.

Концепция свойства, присутствующая в некоторых технологиях (C#, JavaBeans), более удачна. Она сочетает в себе как признаки поля объекта (типизированность, строго заданный набор операций: чтение, запись), так и признаки метода (за операциями может скрываться определённая программистом функциональность). Модель взаимодействия компонентов, основанная на свойствах, проста и формальна в силу ограниченного количества характеристик понятия «свойство» и операций над ним. Однако свойства, как они реализованы, напр., в C#, не могут конкурировать с методами в части предоставляемых разработчику возможностей. Так, если свойство предоставляет лишь доступ для чтения/записи, реализация широко известного механизма обратных вызовов затруднительна. Т.о., мы приходим к необходимости добавления операции *связывания.*

Итак, мы указали на два ключевых принципа — иерархическую организацию компонентов и их взаимодействие, основанное на свойствах, — следование которым, на наш взгляд, необходимо для создания компонентной технологии, удобной в практическом применении. Наряду с требованием возможности определения новых типов компонентов во время исполнения, эти принципы легли в основу предлагаемого решения, к описанию которого мы сейчас переходим.

4. Предлагаемое решение

Охарактеризуем компонентную модель, реализация которой позволит сохранить преимущества существующих технологий и моделей, устранив перечисленные выше ограничения.

К числу основных категорий, которыми оперирует модель, относятся: компонент, тип компонента и редактор типа компонента. Формальные определения приведены ниже, но, забегая вперёд, отметим, что под компонентом понимается сущность, представляющая собой основной «строительный блок» программной системы и инкапсулирующая данные и поведение. Тип компонента — это сущность, описывающая тип данных, хранимых в компонентах данного типа, и содержащая другую метаинформацию. Редактор типа — это средство определения новых типов компонентов на основе существующих.

4.1. Компонент

Идея компонентной разработки программного обеспечения в первую очередь предполагает разграничение «интерфейс — реализация», что позволяет осуществлять разработку и использование компонента независимо.

Под компонентом будем понимать «чёрный ящик», функциональность и данные, скрытые за интерфейсом. Компонент, т.о., характеризуется

интерфейсом и реализацией, а также состоянием, т.е. данными, содержащимися в нём и изменяющимися в течение жизни.

Интерфейс компонента определяется набором его свойств — именованных атрибутов, для которых определён тип значения и права доступа.

Реализация определяет поведение компонента, его функциональность и отличается у *примитивных, составных* и *скомпилированных* компонентов.

Интерфейс

Элементарной ячейкой интерфейсной части компонента является его *свойство*. *Свойством* мы называем имеющую строго определённый тип именованную сущность, представляющую некоторый аспект или атрибут компонента, к которой, возможно, применимы некоторые из трёх операций (*см. рис. 1*):

✦чтение — получение текущего значения данного свойства;

✦ запись — установка текущего значения;

✦ связывание — операция подключения «слушателя» события изменения свойства.

Применимость данных операций определяется правами доступа (устанавливаемыми типом компонента, см. раздел 4.2).

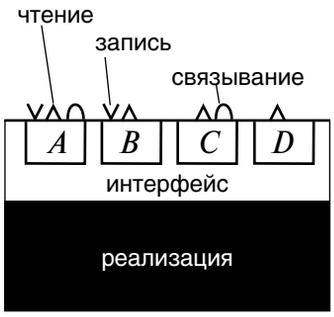


Рис. 1. Интерфейс компонента, состоящий из свойств *A*, *B*, *C* и *D* с различными режимами доступа.

Значение свойства, если оно доступно для чтения, определяется внутренним состоянием компонента, на которое, в свою очередь, влияют операции записи свойств.

Операция связывания свойства A со свойством B (A и B могут принадлежать различным компонентам) позволяет компоненту-владельцу свойства B получать уведомления о каждом изменении значения свойства A в виде записи нового значения в свойство B.

Реализация

Придерживаясь идеи организации компонентов в иерархические структуры, мы различаем три вида компонентов: примитивные, скомпилированные и составные.

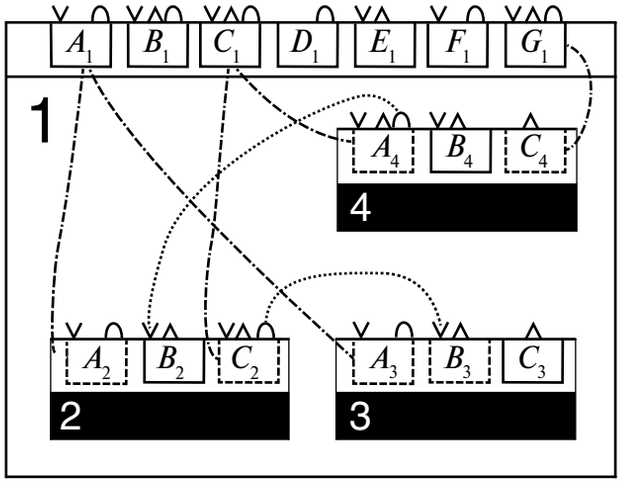


Рис. 2. Реализация: компонент содержит в себе три подкомпонента, связанных друг с другом и со свойствами надкомпонента.

Пунктирные линии показывают установленные событийные связи, штрихпунктирные соединяют между собой разделяемые свойства.

Примитивные компоненты аналогичны примитивным типам в языках программирования. Они инкапсулируют данные наиболее распространённого вида: целые и дробные числа, логические значения, символы, строки символов и ряд других. С точки зрения модели, примитивные компоненты неделимы, т.е. не состоят из других компонентов, и не обладают свойствами.

Примитивные компоненты — типичный пример «объектов-значений», значение которых задаётся при создании и не может быть изменено. Важнейшей характеристикой примитивного компонента является индивидуальность его значения.

Составные компоненты представляют собой (с точки зрения реализации) наборы других компонентов, сочленённых посредством двух механизмов: *событий* и *разделяемых свойств*. В подобных случаях мы будем говорить об объемлющем компоненте (надкомпоненте) и вложенных компонентах (подкомпонентах).

Событийная связь предполагает, что над свойствами двух подкомпонентов была произведена описанная выше операция связывания некоторых из их свойств.

Связь посредством разделения подразумевает следующее. Компонент, создаваемый «внутри» другого компонента, в процессе создания получает ссылку на свойство своего надкомпонента. Это вынуждает его не создавать новое свойство (что делается при создании компонента «вовне» других компонентов), а использовать существующее свойство объемлющего компонента для последующих операций взаимодействия с внешним миром (чтение/запись/связывание). Реализация данного механизма описана в разделе 4.2. Одно свойство надкомпонента может быть разделено несколькими свойствами подкомпонентов, что позволяет достаточно гибко организовывать взаимодействие в системе.

Заметим, что способ «прикрепления» свойства компонента к его реализации (определения, каким образом операции над свойством взаимодействуют с реализацией) в принципе может иметь достаточно сложный характер. Одна крайность – это описание взаимодействия на некотором языке программирования, другая – это тривиальная реализация, когда свойство лишь предоставляет доступ к внутренней переменной компонента. Вышеописанный подход – компромисс, не перегруженный сложностью и не лишённый гибкости.

Подкомпоненты некоторого компонента могут сами являться составными компонентами. Подобная иерархия может иметь рекурсивную вложенность произвольной глубины (ограниченную лишь доступными аппаратными средствами), будучи на нижнем уровне представлена примитивными компонентами.

Скомпилированные компоненты — это компоненты, реализованные средствами, отличными от описанных выше средств реализации составных компонентов. Их поддержка необходима для интеграции со сторонними технологиями (такими как JavaBeans).

Анализ реализации скомпилированного компонента моделью не производится. Этим скомпилированные компоненты схожи с примитивными. Два основных различия: 1) последние не имеют свойств и являются (неизменяемыми) компонентами-значениями; 2) скомпилированные компоненты, как и составные, имеют состояние по умолчанию и могут быть сконструированы вне какого-либо внешнего контекста, тогда как создание примитивных компонентов без указания как минимум их значения лишено смысла.

Перейдём теперь к описанию типов компонентов и механизмов их создания.

4.2. Тип компонента

Тип компонента представляет собой именованную сущность, специфицирующую, каким образом должен создаваться компонент данного типа.[1] Т.о., понятие типа аналогично понятию класса в объектно-ориентированных языках.

Типы содержат в себе описание интерфейса, реализации и их взаимосвязи для компонентов данного типа. «Интерфейсная» часть типа содержит перечень *дескрипторов свойств*, каждый из которых задаёт имя (либо индекс или любой другой «адрес») свойства будущего компонента, его тип, права доступа и, возможно, значение по умолчанию.

«Реализационная» часть типа различна для примитивных, скомпилированных и составных типов (однозначно соответствующих трём рассмотренным категориям компонентов). Для примитивных типов она сводится к хранению текущего значения компонента. Для скомпилированных типов — это указание способа получения реализации создаваемого компонента и связывания его с интерфейсом. Так, для создания компонента, реализация которого основана на технологии JavaBeans, достаточно указания имени соответствующего java-класса — дальнейшие действия осуществляются с помощью механизма отражений Java [6].

Наибольший интерес представляет составной тип. В части реализации он включает в себя список *дескрипторов подкомпонентов*, каждый из которых задаёт тип соответствующего подкомпонента в будущем компоненте и, возможно, его начальное значение.

Эта информация дополняется данными о связях:

✦ для каждого свойства каждого подкомпонента указывается, какое свойство надкомпонента оно разделяет (если разделение имеет место);

✦ перечисляются событийные связи для тех свойств (подкомпонентов и надкомпонента), для которых это необходимо.

Прежде чем переходить к описанию процесса *инстанцирования* (создания экземпляра) типа, введём понятие *поля* компонента. Поля являются средством реализации механизма разделения свойств. Если компонент создаётся вне другого компонента,

[1] Всякий компонент в момент создания получает и сохраняет ссылку на собственный тип, тем самым однозначно ассоциируя себя с ним.

он создаёт «ячейку», соответствующую этому свойству, с которой взаимодействует его внутренняя реализация. Но если компонент в момент инициализации получает ссылку на некоторое свойство надкомпонента, он должен использовать её везде, где в обычном случае использовал бы упомянутую «ячейку». Т.о., эта «ячейка» должна иметь ссылочный тип, исполняя роль посредника во взаимодействии реализации компонента с его свойствами, а адрес ссылки должен определяться контекстом инициализации компонента. Подобную ячейку-ссылку мы и называем *полем*.[2]

Опишем теперь процесс конструирования компонента. Он подразумевает следующую последовательность шагов.

1. Создаются и инициализируются (в зависимости от контекста) поля компонента (по одному на каждый дескриптор свойства составного типа или описание свойства скопилированного типа).

2. Создаётся реализация компонента. В случае составного типа создаются подкомпоненты (по одному на дескриптор подкомпонента); если свойство A_i некоторого подкомпонента i должно быть разделено со свойством A_0 надкомпонента, в момент инициализации компоненту i передаётся ссылка на A_0. В случае примитивного типа в компоненте сохраняется его значение, заданное пользователем. В случае скомпилированного типа создание реализационной части компонента зависит от «нижележащей» технологии.

3. Интерфейс «прикрепляется» к реализации. Для составного компонента это означает установку необходимых событийных связей между его свойствами и свойствами подкомпонентов. Для скомпилированного компонента данный шаг, как и предыдущий, зависит от технологии, на основе которой он реализован.

После инстанциирования типа результирующий компонент обладает соответствующими свойствами и реализацией. Все ограничения, отслеживаемые компонентами в процессе функционирования, – контроль типов, доступа и др. – основываются на метаинформации, заложенной в их типах.

Заметим, что, стремясь минимизировать количество независимых категорий, которыми оперирует модель, мы считаем типы компонентами. Т.о., существует тип «тип», экземплярами которого являются все типы (в т.ч. и сам этот тип). Перечисленное выше «содержимое» типа (перечни дескрипторов свойств, событийных связей, дескрипторов подкомпонентов) — это три свойства типа «тип».

Специфика понятия «тип» накладывает важное ограничение на манипулирование типами как компонентами: тип по своей сути является сущностью неизменяемой, т.к. содержит в себе метаинформацию о других сущностях. Необходимость определения новых типов во время исполнения, о которой шла речь выше, не требует, тем не менее, модификации существующих типов: достаточно иметь возможность создания нового типа, являющегося модифицированной версией прежнего.

4.3. Определение типов во время исполнения

Рассмотрим вопрос определения новых типов во время исполнения. Предлагаемая модель предоставляет возможность следующей логики действий.

1. Пользователь, выбрав тип компонента из библиотеки типов (включающей в себя предопределённые примитивные типы, а также составные типы, определённые ранее), создаёт *редактор типа* — специальную сущность, в которую заключена функциональность определения новых типов на основе существующих. Она принимает на вход существующий тип и позволяет совершать операции, перечисленные ниже.

2. Пользователь имеет возможность настроить полученную структуру.

- ✦ Он может изменить дескрипторы свойств: имя, значение по умолчанию, права доступа. Важно, что компонент «глубоко» подстраивается под контекст. Так, если пользователь указал, что свойство, прежде доступное для чтения, записи и связывания, будет доступно только для чтения, реализация свойства как переменной должна измениться на реализацию в виде константы (в смысле языка программирования).
- ✦ Пользователь может редактировать метаданные реализации: добавлять и удалять подкомпоненты, устанавливать и удалять событийные связи между ними, добавлять и удалять отношения разделения свойств.

3. Когда необходимые модификации произведены, созданный составной тип можно добавить в

[2] Описанное понятие сходно с концепцией поля в языке VRML [7].

библиотеку типов и в дальнейшем использовать его наравне с другими типами.

Редактор типа, т.о., предоставляет доступ к внутренним данным типа и позволяет модифицировать их *копию*, с тем чтобы иметь возможность создавать видоизменённые варианты типов, экземпляры которых существуют в системе. Реализация редактора типа, благодаря такому подходу, становится достаточно простой; мы не станем останавливаться на ней в данной работе.

В заключение заметим также, что редактор типа позволяет помещать произвольную редактируемую им структуру в составной компонент. Это необходимо для полноты модели, в частности, на начальном этапе, когда в библиотеке имеются только примитивные типы.

5. Заключение

Мы рассмотрели основные положения новой компонентной архитектуры, расширяющей возможности существующих компонентных моделей. Мы полагаем (и постарались аргументировать эту точку зрения), что принципы, заложенные в ней, обусловливают высокую гибкость и универсальность. Возможность порождения новых типов позволит системам, реализованным на базе данной архитектуры, развиваться значительно более интенсивно, чем это доступно традиционному программному обеспечению. ■

Литература

1. Boudreau T., Glick J., Greene S. NetBeans: The Definitive Guide. — O'Reilly Media, 2002. — 672 с.
2. Bruneton E., Coupaye T., Stefani J.B. The Fractal Component Model specification. Version 2.0-3. — The ObjectWeb Consortium, 2004. — 52 с.
3. Costa Seco J., Silva R., Piriquito M. ComponentJ: A Component-Based Programming Language with Dynamic Reconfiguration. // Computer Science and Information System. —2008. — Vol. 5, No. 2. — С. 63-86.
4. Dos Reis G., Järvi J. What is Generic Programming? // Library-Centric Software Design. —2005. — С. 1-11.
5. Gamma E., Helm R., Johnson R. Design Patterns: Elements of Reusable Object-Oriented Software. — Addison-Wesley, 1994. — 416 с.
6. Gosling J., Joy B., Steele G. The Java™ Language Specification. — 3-е изд. —Addison Wesley, 2005. — 688 с.
7. International Standard ISO/IEC 14772-1:1997. The Virtual Reality Modeling Language. URL: http://www.web3d.org/x3d/specifications/vrml/ISO-IEC-14772-VRML97/ (дата обращения: 20.05.2010)
8. Krueger C.W. Software reuse. // ACM Comput. Surv. — Vol. 2. — New York: ACM. — 1992. — С. 131-183.
9. McIlroy M.D. Mass produced software components. // Naur P., Randell B., «Software Engineering, Report on a conference sponsored by the NATO Science Committee, Garmisch, Germany, 7th to 11th October 1968». — Brussels: Scientific Affairs Division, NATO. — 1969. — С. 138-155.
10. Object Management Group. The Common Object Request Broker: Architecture and Specification. Version 3.1. Part 3 - Components. — OMG document formal/2008-01-08. — 2008. — 192 с.
11. Redmond F.E. DCOM: Microsoft Distributed Component Object Model. — Foster City: IDG Books Worldwide, Inc. — 1997. — 384 с.
12. Stiemerling O. Component-Based Tailorability. — Bonn: Bonn University. — 2000. — 180 с.
13. Sun Microsystems Inc. The JavaBeans™ API specification. Version 1.01-A. — Sun Microsystems Inc. — 1997. — 114 с.

ИНСТРУМЕНТАЛЬНАЯ РЕАЛИЗАЦИЯ АРХИТЕКТУРНЫХ МОДЕЛЕЙ ПРЕДПРИЯТИЯ НА ОСНОВЕ ОНТОЛОГИЙ

Н.Н. Лычкина,
кандидат экономических наук, доцент, заместитель заведующего кафедрой «Информационные системы» Государственного университета управления,
e-mail: lychkina@guu.ru.

А.Р. Идиатуллин,
аспирант кафедры «Информационные системы» Государственного университета управления,
e-mail: idiatulla@gmail.com.
Адрес: г. Москва, Рязанский проспект, д. 99.

В статье рассматриваются вопросы сквозного сервисно-ориентированного проектирования интегрированных информационно-аналитических систем предприятия на основе архитектурного подхода с применением онтологических моделей архитектуры предприятия. Приводится описание комплекса онтологических моделей предметной области предприятия: мета-онтологии, модели бизнес-среды и онтологии процедур стратегических информационных систем, соответствующих различным уровням рамочной схемы архитектуры предприятия.

Ключевые слова: информационная система предприятия, система поддержки принятия решений, онтологии, семантическая сеть моделей, архитектура предприятия, рамочная схема архитектуры предприятия.

Введение

В основе теоретических и методологических подходов к управлению информационной системой предприятия лежит концептуальное представление организации как функционирующей бизнес-системы в условиях динамично изменяющейся внешней среды, что на практике требует обеспечения сопоставимости и преемственности бизнес-стратегии и ИТ-стратегии предприятия на основе моделей стратегического соответствия, постоянного реинжиниринга предприятия и его информационной системы, а также реализацию ИТ-стратегии через ИТ-инфраструктуру на всех уровнях управления. Эффективное планирование и реализация ИТ-стратегии сегодня связывают со *сквозным сервисно-ориентированным проектированием (ССП)* на основе архитектурного подхода.

ССП позволяет выстраивать целостный подход к проектированию корпоративных информационных систем – от предприятия в целом и выделения бизнес-сервисов на верхних уровнях архитектуры до его информационной системы и отображения бизнес-сервисов в прикладные ИТ-сервисы на технологическом уровне.

Фундамент для полноценного сквозного сервисно-ориентированного проектирования обеспечивает архитектурный подход и специальные методы и модели анализа и синтеза сервисной бизнес-архитектуры и ее информационной инфраструктуры, фактическим стандартом которого явилась модель Джона Захмана. Концепция Архитектуры предприятия (EA) является способом объединения и синхронизации функциональных и бизнес-потребностей организаций с возможностями информационных технологий в условиях их возрастающей сложности [1]. Таким образом, модель Захмана может рассматриваться в качестве общей рамочной схемы описания взаимосвязи сервисов на различных уровнях архитектуры, при выборе отдельных архитектурных решений и описания элементов этой архитектуры с определенными уровнями абстракции и агрегации.

1. Перспективные методологии и инструменты архитектуры предприятия и моделирования предприятия, бизнес-процессов и процессов управления развитием предприятия

Организации все больше проникаются осознанием необходимости применения стандартов и инструментов EA. Расширение и унификация инструментария в области EA позволяет преодолевать традиционные видения и сложившиеся стереотипы мышления как у бизнеса, так и у системных архитекторов. Устройство общих схем (frameworks) EA достаточно сложно, полная схема EA по своей сути представляет многоуровневые иерархические структуры типа страт или эшелонов, с которыми работает бизнес-архитектор и системный архитектор, причем работа с объектами является достаточно сложной. Выстраивание целостного сервис-ориентированного подхода к проектированию архитектуры предприятия и его информационной системы требует совершенствования методов и инструментария архитектурной работы на основе рамочной схемы EA. При навигации по уровням архитектуры предприятия необходимо поддерживать в процессе проектирования единый формат представления информации и обеспечивать взаимодействие и совместную работу бизнес-архитекторов и системных архитекторов. В части архитектурной работы с бизнес-процессами сегодня широко используются методологии и технологии моделирования бизнес-процессов, языки организационного моделирования, поддерживающие автоматизацию исполняемых бизнес-процессов на технологическом уровне рамочной схемы архитектуры предприятия.

За последнее время дисциплина EA прошла определенную эволюцию в своем развитии, предложила ряд референсных моделей. Все больше организаций определяют собственную «рамочную» (framework) схему EA вместо использования или простой адаптации существующих схем. Однако на практике проектирование архитектур предприятий не всегда осуществляется с использованием специализированной инструментальной поддержки. Существующие инструментальные средства моделирования архитектуры предприятия не обладают достаточной функциональностью. Инструментальные решения в этой области включают универсальные интегрирующие среды (например, Zachman Framework, GERAM), языки моделирования предприятий (например, IDEF, ARIS, BPML), программные среды моделирования (например, ARIS 6 Collaborative Suite, Popkin System Architect, METIS), мета-модели и языки мета-моделирования (например, UML Profile for Business Process Definition, UEML, Essential Project) и др. Исследование инструментальных средств моделирования EA позволило выявить их недостатки. Так Zachman Framework не поддерживает представление динамики развития организации и ее информационных систем (отсутствие оси времени), является достаточно поверхностной с точки зрения детализации референсной моделью, недостаточно функциональна с технической точки зрения. GERAM носит концептуальный характер, она предоставляет методологические руководства, но не обеспечивает ни языком моделирования, ни соответствующими инструментальными средствами. ARIS призвана поддерживать общий репозиторий однотипных объектов различных методологий, назначать объектом одних методологий модели других методологий. Однако, несмотря на обилие графических представлений, она не позволяет отображать некоторые виды моделей, необходимые для архитектурной работы. Это позволяет сделать вывод о необходимости разработки единой ком-

плексной инструментальной среды моделирования архитектуры предприятия и его информационной системы.

Анализируя модели и методы стратегического менеджмента такие как SWOT-анализ, GAP-анализ, матрица BCG и PEST-анализ, их место в построении архитектуры предприятия и его информационной системы, можно сказать, что не все из них поддержаны инструментальными средствами моделирования, они сфокусированы на определенном аспекте архитектуры предприятия и, как следствие, слабо ориентированы на комплексное всестороннее описание архитектуры предприятия.

Исследование методологий проектирования информационных систем, такие как RUP (поддержанная UML), SADT, BPML, DFD и семейство IDEFx, в контексте описания архитектуры предприятия, показывает, что данные методологии обладают рядом свойств, ограничивающих их применение: они предназначены прежде всего, для описания состава структурированных исполняемых процессов, протекающих в рассматриваемой системе, а не их последовательности, узко-специализированы, и зачастую включают избыточные или практически неиспользуемые диаграммы и конструкции, обладающие неточной семантикой. Большое количество различных методологий и технологических решений, их поддерживающих, множественность языков организационного моделирования порождает рассогласованность моделей и так называемую проблему «вавилонской башни» [2], что затрудняет их освоение менеджерами и применение на практике. Множественность языков представления предметной области ставит задачу согласования моделей, описанных с помощью различных языков и методологий и поиска единого языка описания предметной области предприятия. Кроме того, существующие методологии и технологии проектирования информационных систем хорошо справляются с функцией автоматизации четко формализуемых и структурированных исполняемых процессов на уровне транзакционных систем и, как показывает анализ, слабо приспособлены для описания связанных аналитических приложений в информационно-аналитических системах, где основой автоматизации является процесс принятия решения, отличающийся нерегулярностью и существенной нелинейностью.

Необходима интегрированная компьютерная среда, поддерживающая как традиционные функции моделирования бизнес-процессов, так и архитектурную работу на более высоких уровнях концептуализации предметной области предприятия, что позволит обнаруживать точки пересечения и взаимосвязи между бизнес-функциями и бизнес-стратегией, а также ИТ-ресурсами. В перспективе с помощью такой интегрированной программной среды EA возможно будет осуществлять не только документирование бизнес-процессов компании, но и каталогизацию всей корпоративной базы знаний. Задача состояла в том, чтобы разработать такой комплексный инструментарий для архитектурной работы на высоких уровнях концептуализации для интеграции различных моделей предметной области предприятия, где рамочная схема EA выступает в качестве исходной методологической основы.

Анализ методик и моделей описания EA и инструментов и технологий моделирования EA показал, что необходимы инструменты, модели и методы для общего описания организации и проектируемой информационной системы на основе рамочных схем (FW EA), отвечающие следующим требованиям:

✦ комплексное описание предметной области организации и ее информационной системы с возможностью интеграции и анализа различных архитектурных аспектов,

✦ содержание наиболее общих и основополагающих понятий и отношений в моделируемой предметной области организации с возможностью детализации различных архитектурных аспектов,

✦ возможность описания архитектуры предприятия на основе рамочных схем,

✦ навигация по различным уровням архитектурных схем организации, включая уровень разработки приложений и данных информационной системы организации,

✦ машинно-читаемость и возможность трансляции на другие языки моделирования целиком или частично, для интеграции с существующими методологиями и инструментальными средствами проектирования информационной системы организации,

✦ обеспечение инструментами визуализации моделей EA, с возможностью их интерпретации широким кругом специалистов;

✦ обеспечение единства понятийного аппарата и создание глоссария для всех участников процесса проектирования информационной системы организации.

В качестве метода комплексного описания архитектуры предприятия предложен онтологический инжиниринг.

В работе [3] высказывается предположение о применение данного метода в качестве инструмента моделирования бизнес-процессов, где обобщенная метамодель предприятии в виде онтологии может служить основой для интеграции разнородных представлений бизнес-процессов в корпоративной информационной системе. Кроме того, в области проектирования архитектуры предприятия были попытки применения онтологий, к примеру, в инструменте Essential Project сделана попытка создания комплексной метамодели предприятия на языке онтологий.

Метод онтологического моделирования был исследован в контексте применимости для описания архитектуры предприятия, что позволило сделать выводы о том, что он обеспечивает комплексное описание предметной области организации и ее информационной системы, позволяет моделировать содержание наиболее общих и основополагающих понятий и отношений предметной области организации, при этом он не ограничивается жесткой методологией применения, что дает возможность описания архитектуры предприятия на основе различных рамочных схем. Метод инженерии онтологий поддерживается широким набором инструментальных средств, что позволяет создавать машинно-читаемые модели, визуализировать их, обеспечивая навигацию по различным уровням архитектурных схем организации и оставляя возможность для интеграции с другими языками моделирования.

Инструментальные средства на основе онтологий естественным образом стыкуются с другими видами схем стратегического и архитектурного уровней. К таким схемам относятся, например, схемы циклов маркетингового стратегического управления, или такие схемы создания ИС и ИУС, схемы бизнес- и информационной платформ и архитектур Дж. Хендерсона, схемы «здания ARIS» А. Шеера. А благодаря концептуальной совместимости с другими схемами после описания целостной и динамичной 3D-архитектуры можно легко включать в работу более специфические или более технические инструменты и модели, например, Turbo-BPR, Process Charter, ARIS ToolSet, UML RRose или Oracle Designer.

Применение онтологического подхода при формализации базовых категорий предметной области предприятия обладает следующими преимуществами:

✧ Модель универсальна и способна описать различные аспекты корпоративной архитектуры – от системы стратегий и целей до организационной структуры и системы бизнес-процессов.

✧ Применима на различных уровнях детализации – от верхнего уровня описания базовых категорий менеджмента, до уровня проектирования аналитических приложений.

✧ Легко адаптируема и дополняема, при этом глубина проработки отдельных аспектов определяется практической необходимостью и не регламентирована методологией.

✧ Отдельные аспекты архитектуры предприятия могут быть описаны на других языках моделирования, с использованием понятий, введенных в онтологической модели.

✧ Комплекс онтологических моделей позволяет выстраивать взаимосвязи сервисов и приложений на разных уровнях архитектуры, за счет визуализации упрощает восприятие и работу со сложными архитектурными схемами с большим количеством объектов во взаимосвязанных измерениях обобщения и агрегации и детализации на разных концептуальных уровнях представления предприятия.

✧ Машинно-читаемость и транслируемость онтологической модели в другие универсальные языки.

На основе онтологических моделей создана комплексная инструментальная среда моделирования архитектуры предприятия и его информационной системы. Выделены классификационные признаки создаваемых онтологий предметной области организации: по области применения онтологии относятся к моделированию и проектированию, выполняют роль – интеграция разнородной информации, владельцем онтологий является организация, язык – неформальный, предметной областью является организация, глубина проработки может быть различной в зависимости от решаемых задач. Для покрытия различных аспектов описания предметной области предприятия создано несколько различных по уровню обобщения моделей.

В статье рассматривается применение метода онтологического инжиниринга для формализации и моделирования предметной области предприятия и его информационной системы на основе рамочной схемы архитектуры предприятия.

2. Комплекс онтологических моделей архитектуры предприятия

Разработанный комплекс архитектурных моделей предприятия представлен в виде семантической сети онтологий. Различные модели семантической сети соответствуют различным осям и аспектам рамочной схемы архитектуры предприятия. Комплекс моделей на основе онтологий отражает схему взаимосвязей между отдельными элементами архитектуры предприятий и используется при формировании и выборе архитектурных решений и элементов, а также для установления горизонтальных и вертикальных связей между ними и различными архитектурными слоями рамочной схемы.

В качестве инструмента инженерии онтологий использован программный продукт Protege 3.4.4.[4].

Процесс создания онтологий включает разработку комплекса моделей различного уровня обобщения, начиная с мета-онтологии, детализируемой с помощью моделей предметной области и прикладных онтологий. Обобщенная онтологическая модель архитектуры предприятия на основе мета-онтологии представлена в виде семантической сети онтологий предметной области, прикладных онтологий и других онтологических моделей процедур, приложений и моделей бизнес-процессов и служит основой для интеграции разнородных представлений в корпоративной информационной системе.

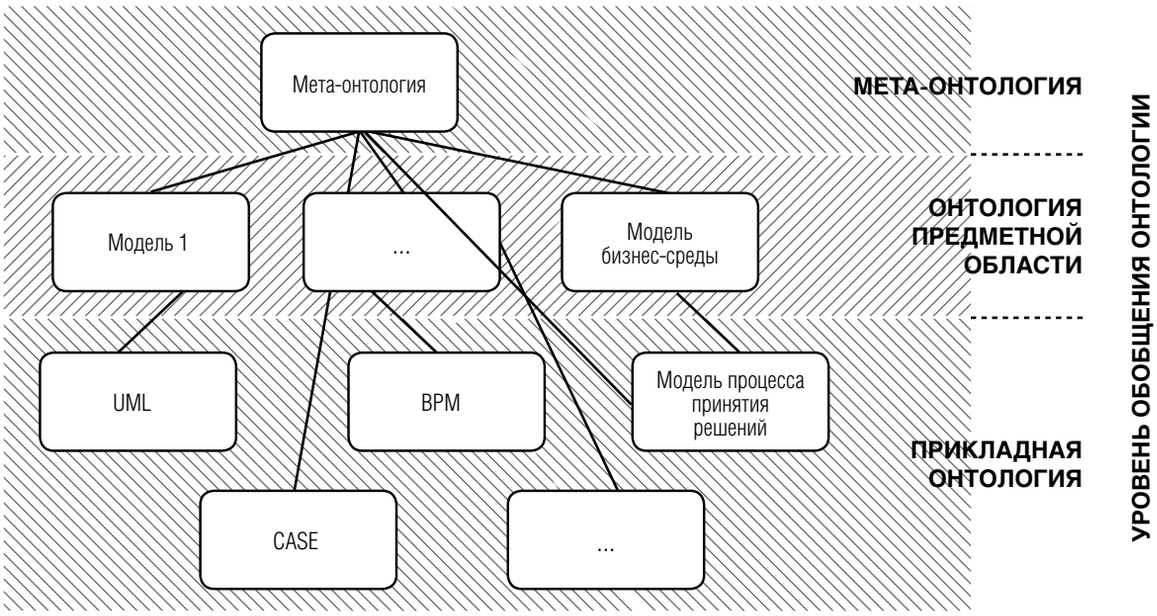


Рис. 1. Семантическая сеть моделей на основе онтологий.

Мета-онтология корпоративной архитектуры

Для координации и навигации по таким уровням архитектуры предприятия, как прикладной, системный и технический, применена обобщающая онтологическая модель, содержащая наиболее общие понятия предметной области организации. Онтологическая модель, описывает общую схему архитектуры предприятия на верхнем концептуальном уровне, называемую мета-онтологией, служит навигатором по семантической сети и является инструментом сквозного сервисного проектирования информационной системы организации. Мета-онтология формализует и описывает взаимосвязь понятий архитектуры организации, от верхних концептуальных слоев (бизнес-среды), до системного уровня, отражающего основные аспекты информационной инфраструктуры.

В качестве теоретической основы создания мета-онтологии использовано вербальное описание архитектуры предприятия в соответствии с концепцией «Корпоративной архитектуры» [5]. Модель содержит такие классы понятий как система целей и стратегий, организационная структура, функцио-

нальная структура, система процессов и система показателей эффективности. Модель детализирует данные понятия и описывает их взаимосвязь с такими элементами архитектуры предприятия как информационная архитектура, техническая архитектура и архитектура знаний. С помощью их взаимодействия отслеживается определяющее влияние стратегии развития предприятия на ИТ-стратегию и ИТ-инфраструктуру организации.

В модели выделены следующие понятия предметной области, реализованные в виде базовых классов онтологической модели:

- ✦ Бизнес-архитектура
- ✦ Информационная архитектура
- ✦ Корпоративное управление
- ✦ Стратегическое управление
- ✦ Операционная модель

Производя исследования для онтологического инжиниринга, был сделан вывод, что бизнес-архитектура состоит из большого количества понятий, и является одной из ключевых составляющих архитектуры предприятия. Рассматривая её составляющие, можно выделить следующие понятия:

- ✧ Архитектура знаний
- ✧ Архитектура человеческих ресурсов
- ✧ Техническая архитектура
- ✧ Система процессов
- ✧ Система показателей эффективности
- ✧ Организационная структура
- ✧ Функциональные задачи
- ✧ Система целей и стратегий

Данные понятия были отражены в модели соответствующими классами.

В качестве класса мета-онтологии, бизнес-архитектура состоит из большого количества подклассов и является одной из ключевых составляющих модели архитектуры предприятия. Рассматривая её составляющие, можно выделить следующие элементы, представленные в виде подклассов в онтологии, демонстрируемые на *рис. 2.*

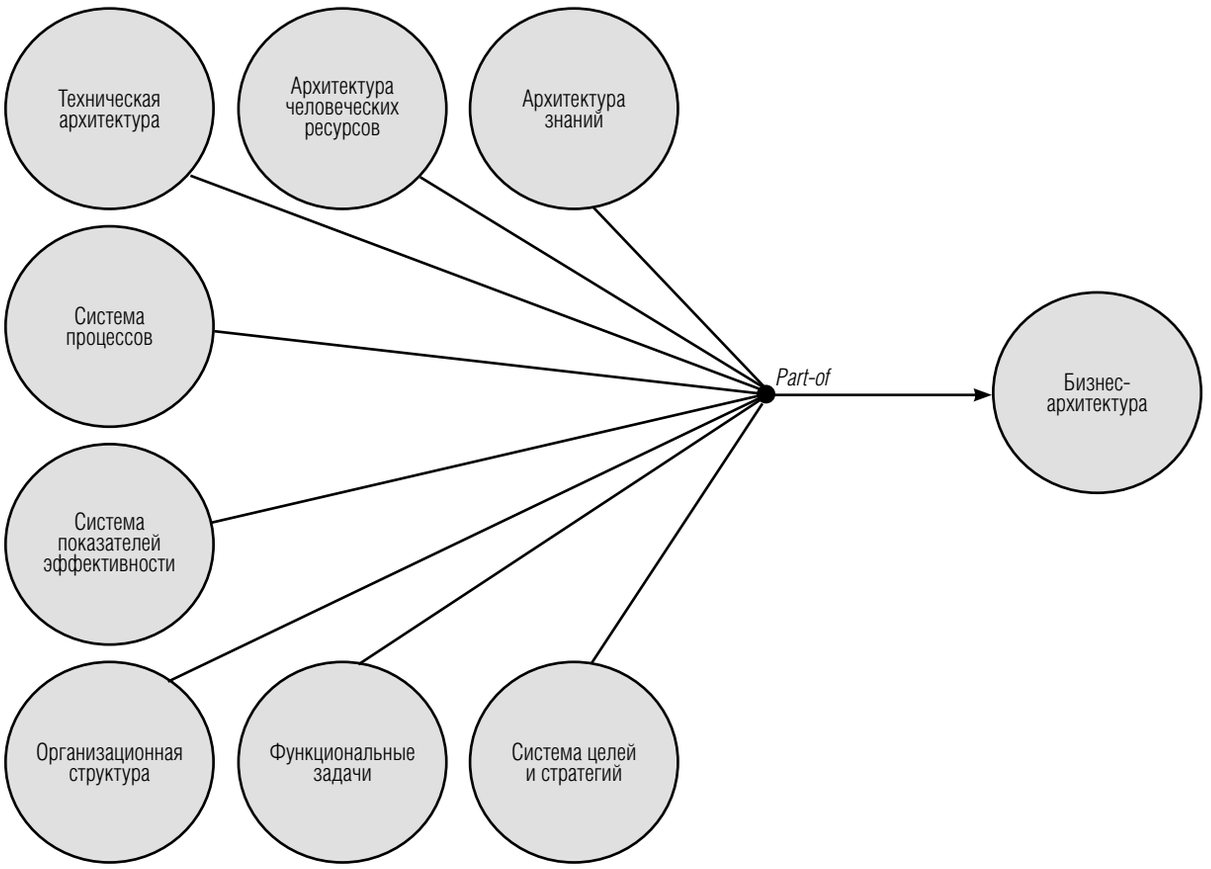


Рис. 2. Композиционная схема класса «Бизнес-архитектура».

Составные классы были так же детализированы в соответствии с описанием архитектуры предприятия. Аналогичным образом в соответствии с вербальным описанием концепции были детализированы и другие базовые классы. Система процессов компании представлена в виде иерархии классов, описывающих пятиуровневую модель процессов организации. Система показателей эффективности раскрывается через различные системы показателей эффективности, такие как система сбалансированных показателей BSC. Операционная модель в рамках онтологии детализируется на такие подклассы как Бизнес-деятельность, Ресурсное обеспечение и Операционное управление, в котором в свою очередь с помощью иерархии классов раскрывается организационная и функциональная структура организации. Выделяют следующие понятия информационной архитектуры:

- Архитектура данных,
- Архитектура приложений и классов,
- Системная ИТ-архитектура.

Рис. 3. Композиционная схема класса «Информационная архитектура».

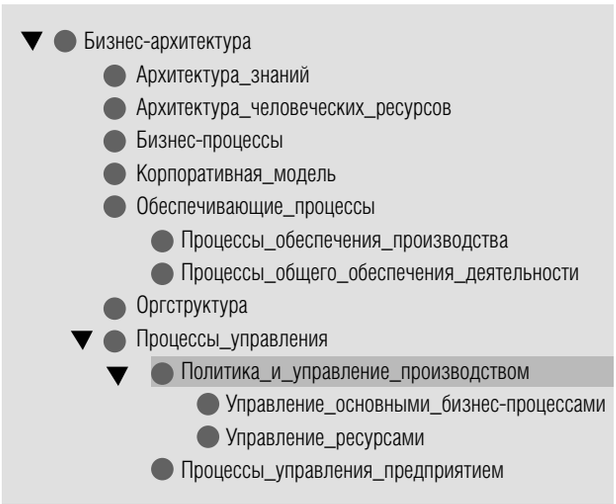


Рис. 4. Фрагмент иерархии классов онтологической модели.

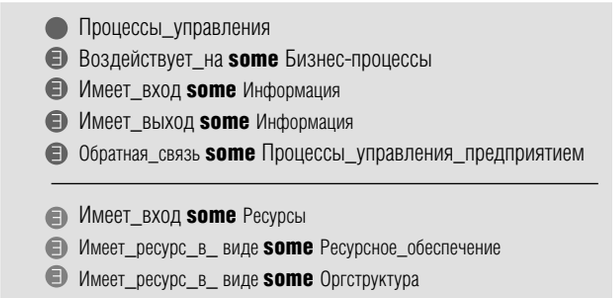


Рис. 5. Значение слотов класса «Управление производством».

Следуя данной логике, были выделены подклассы онтологической модели для класса «Информационная архитектура». В результате в онтологии класс «Информационная архитектура» состоит из следующих подклассов, представленных на *рис. 3:*

На *рис. 4* представлен фрагмент иерархии классов онтологической модели при ее инструментальной реализации, основанный на детализируемых понятиях предметной области организации.

Выделены свойства каждого из классов онтологии и описаны взаимосвязи классов, на основе вербального описания архитектуры организации. На *рис. 5* представлены слоты одного из классов модели, отражающие взаимосвязи понятий иерархии классов.

С помощью возможностей инструмента визуализации можно в наглядном виде представить архитектурные схемы. Инструмент позволяет исследовать архитектурную модель с различных аспектов. К примеру, изучить влияние элементов архитектуры предприятия на исследуемый элемент, либо отследить взаимосвязь элементов с помощью диаграммы взаимосвязи классов, как показано на рисунке 6 на примере взаимодействия элементов системы сбалансированных показателей, отраженных в модели. Таким образом, в рамках модели отражена когнитивная карта системы, но не фрагментарно, а в интеграции с прочими элементами архитектуры предприятия. Благодаря

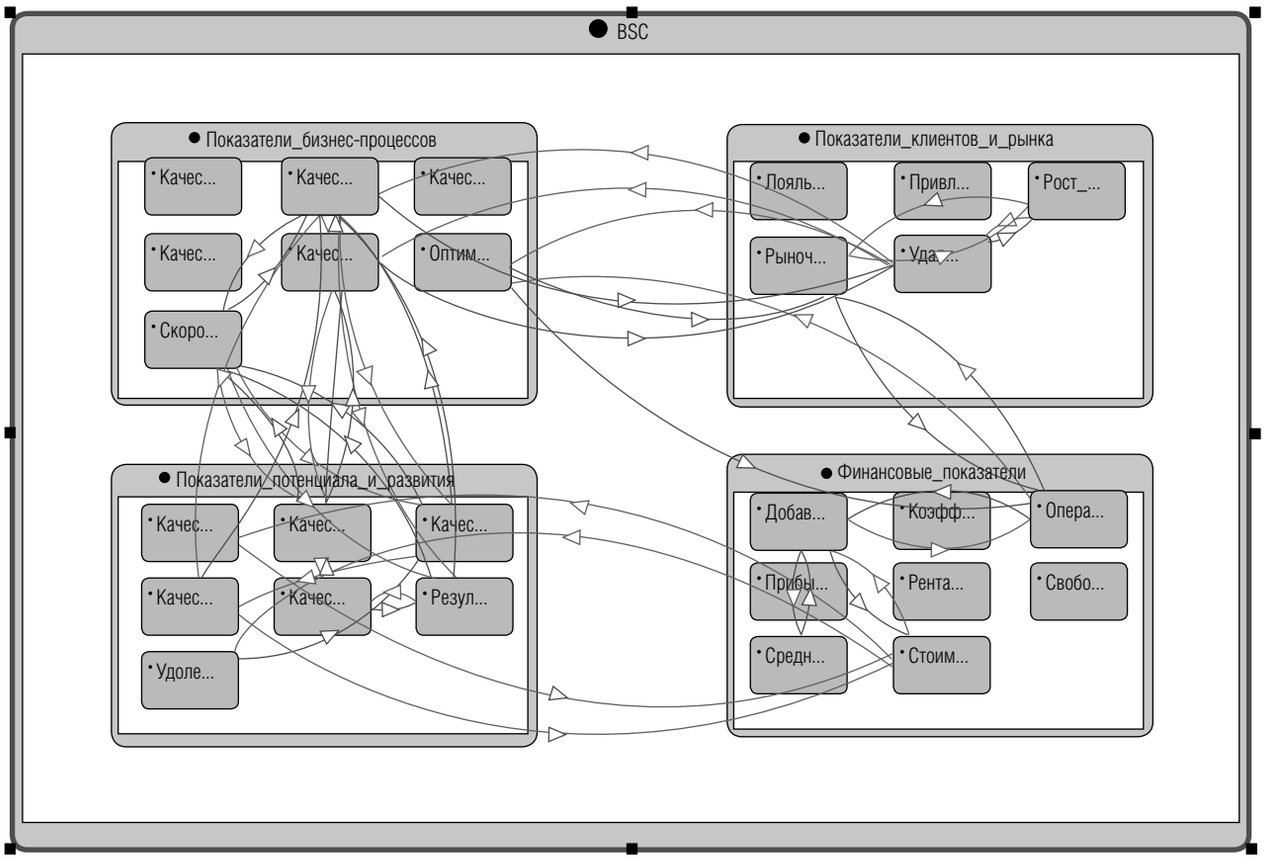


Рис. 6. Фрагмент диаграммы взаимосвязи классов мета-онтологии.

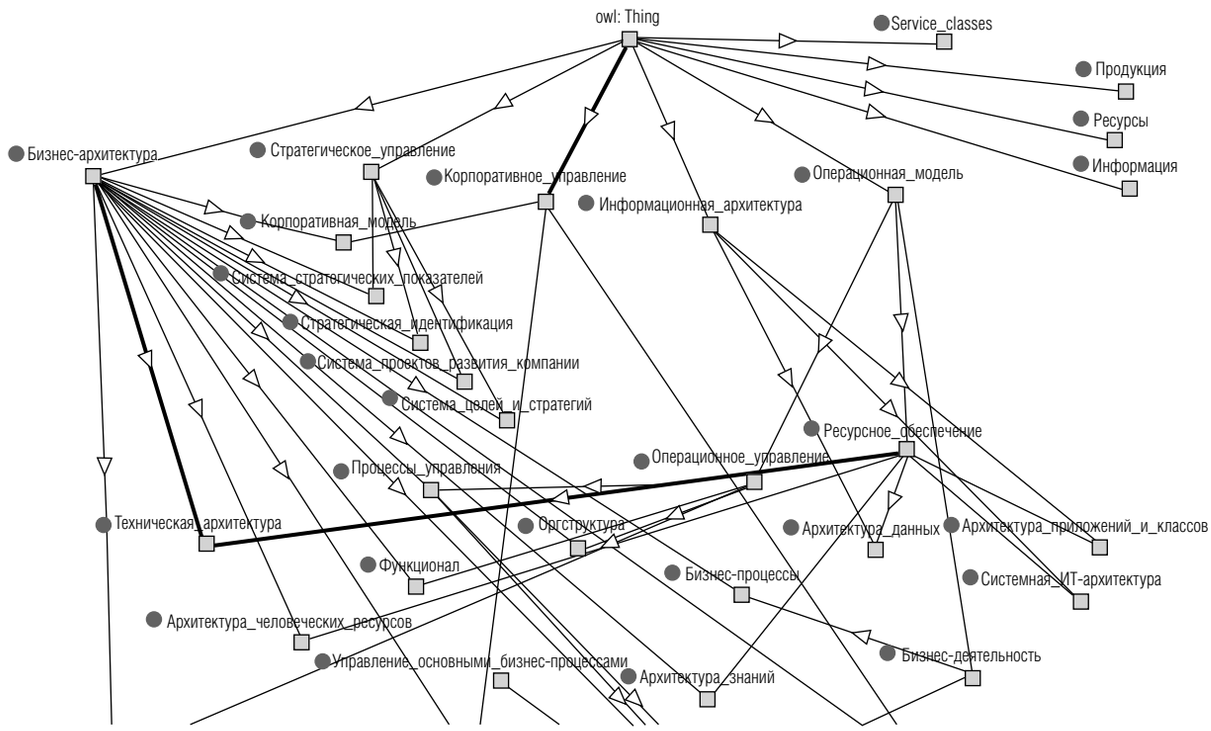


Рис. 7. Фрагмент графа классов мета-онтологии.

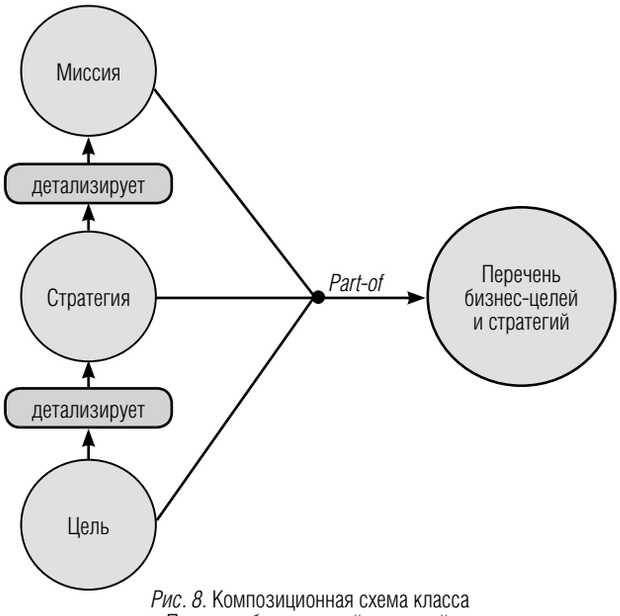


Рис. 8. Композиционная схема класса «Перечень бизнес целей стратегий».

этому можно отследить факторы, влияющие на тот или иной показатель, либо проанализировать, как введение в систему нового показателя эффективности повлечет изменение информационной системы организации.

Визуализация позволяет определить место элемента в иерархии архитектуры предприятия, как показано на *рис. 7.* На схеме представлен фрагмент графа, отражающего общую структуру онтологической модели фреймвока архитектуры предприятия. Рассмотрим элемент «Техническая архитектура» (определен жирными дугами); – на графе видно, что он имеет двойную подчиненность: являясь частью бизнес-архитектуры он одновременно входит в состав ресурсного обеспечения организации как части операционной модели.

Реализованная таким образом модель применима для дальнейшего использования в других системах и инструментах благодаря возможности трансляции в общепринятый язык XML.

Модель охватывает различные аспекты архитектуры предприятия и в процессе проектирования производится формализация стратегии предприятия и преобразование её в ИТ-стратегию. ИТ-стратегия отражается как перечень приложений информационной системы, данных, сущностей, ключевых показателей эффективности, пользователей и системы процессов.

Онтология бизнес-среды организации

На уровне онтологий предметной области некоторые понятия мета-онтология детализируются с помощью модели бизнес-среды организации. Модель детализирует часть понятий мета-онтологии, относящихся к верхнему уровню рамочной концепции архитектуры предприятия. В качестве теоретической основы создания мета-онтологии использовано описание верхнего уровня архитектуры предприятия в соответствии с концепцией Джона Захмана. Онтология бизнас-среды классифицирует следующие обобщенные группы понятий, описанные в концепции: бизнес-мероприятия, система бизнес-целей и стратегий, перечень важных факторов предприятия, организационные единицы,

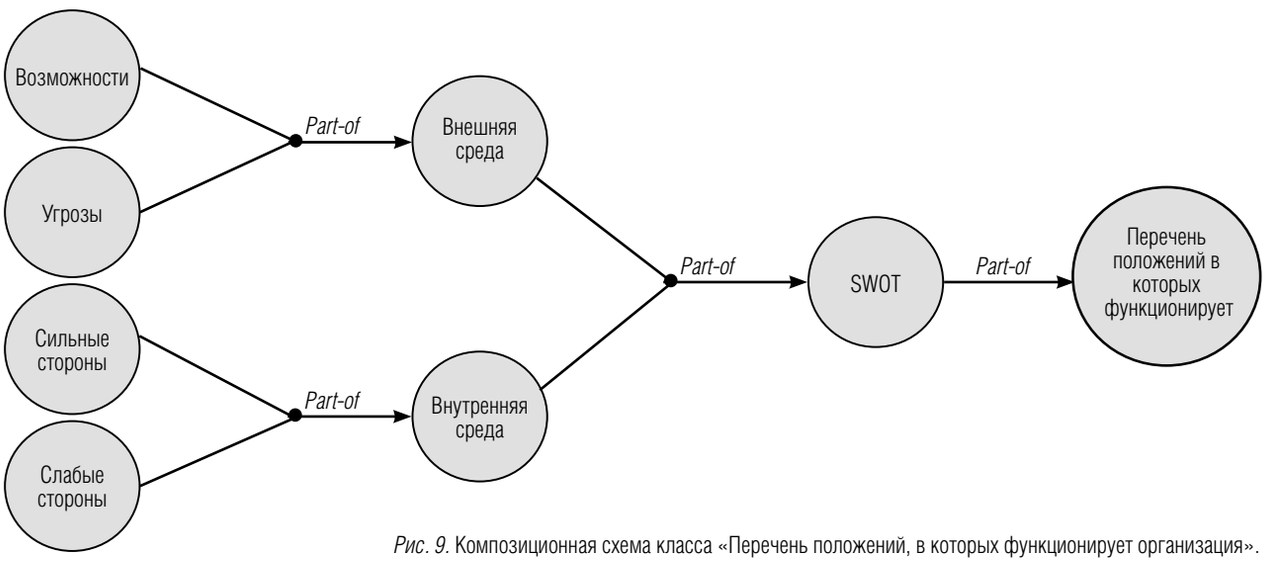


Рис. 9. Композиционная схема класса «Перечень положений, в которых функционирует организация».

перечень положений в которых предприятие функционирует и система процессов предприятия. Данные понятия были выделены в качестве базовых классов онтологии.

Определяющим управляющим воздействием в организации является система целей и стратегий. Таким образом, понятия «Миссия», «Стратегия» и «Цель» являются подклассами понятия «Перечень бизнес целей и стратегий». На языке онтологий это можно выразить с помощью композиционной схемы, представленной на *рис. 8*:

Другие составляющие бизнес-среды были декомпозированы в ходе онтологического ижиниринга с помощью различных моделей менеджмента. Модель включает организационную и функциональную структуру предприятия. Выделен метод стратегического менеджмента, используемый для детализации понятий данного класса – SWOT-анализ. В контексте онтологий – данный метод можно рассмотреть как класс, входящий в состав класса «Перечень положений, в которых функционирует организация» состоящий из подклассов. Перечень его подклассов и их иерархия очевидны из вербальных описаний метода. На *рис. 9* представлена композиционная схема класса «Перечень положений, в которых функционирует организация».

На *рис. 10* представлена инструментальная реализация базовых классов.

Модель оперирует понятиями, имеющими проекцию в мета-онтологии корпоративной архитектуры, в которой система целей и стратегий является подклассом «Стратегическое управление».

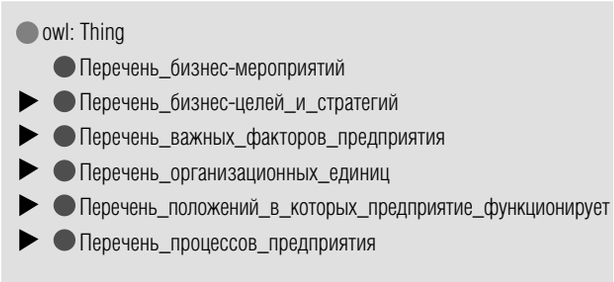


Рис.10. Базовые классы онтологии Бизнес-среды.

Для детализированных подклассов определены слоты, определяющие взаимосвязь с прочими классами. Таким образом, создавая экземпляр какого-либо класса, определяются все взаимосвязи и изменения архитектуры предприятия, которые повлечет исходное действие.

Визуализация модели позволяет в наглядной форме демонстрировать взаимосвязи понятий наименее формализованной предметной области – стратегического управления организацией. На *рис. 11* представлена диаграмма взаимосвязей классов модели. К примеру, на схеме отражены взаимосвязи подклассов класса «Перечень процессов предприятия» и показано, что система процессов должна быть вписана в систему целей, поддержана организационной

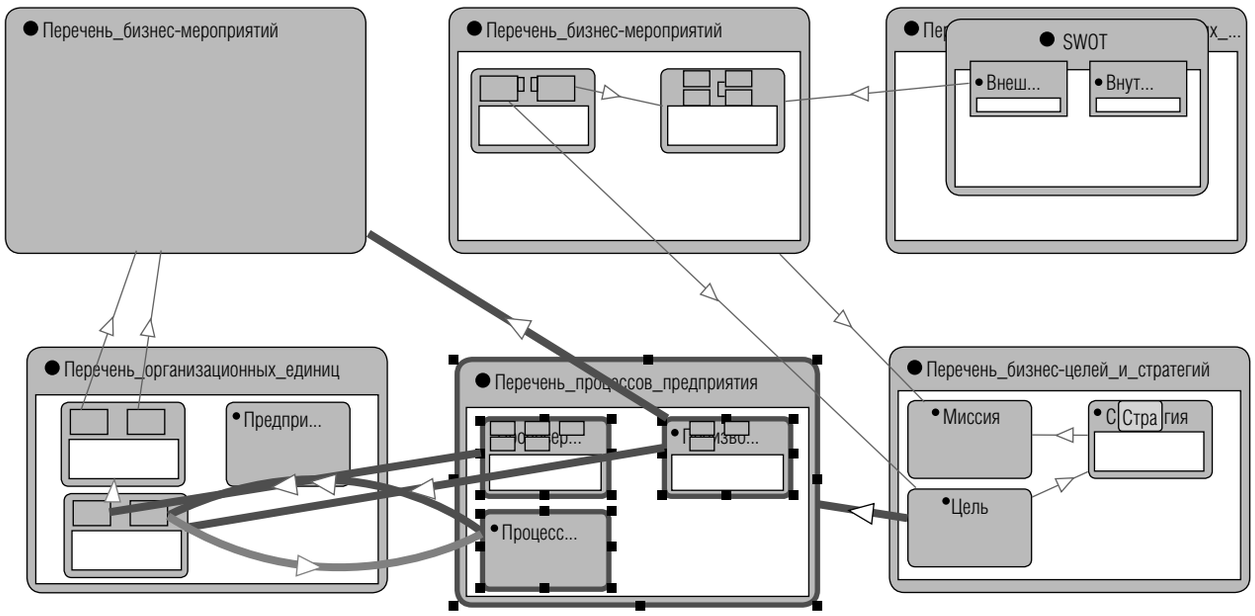


Рис. 11. Диаграмма классов онтологической модели Бизнес-среды предприятия.

структурой и интегрирована с перечнем бизнес-мероприятий (см. жирные стрелки):

Модель предназначена для использования в первую очередь менеджерами компании. Целевым назначением модели является формулировка стратегии предприятия и формализация её в терминах аспектов рамочной схемы архитектуры предприятия. Детализированная с помощью данной модели стратегия ляжет в основу формирования ИТ-стратегии с помощью мета-онтологии.

На нижнем уровне семантической сети могут быть созданы онтологии конкретных приложений, либо модели на других языках организационного моделирования. Для проектирования аналитических приложений в системах управления эффективности бизнеса разработана онтологическая модель процедур поддержки принятия стратегических решений. Модель взаимосвязана с моделями мета-онтологии и бизнес-среды и детализирует понятия стратегии и системы сбалансированных показателей. Прикладная онтология может быть использована для разработки связанных аналитических приложений в BPM-системах, основанных на цикле стратегического управления.

Модель нижнего уровня семантической сети соответствует системному уровню архитектуры предприятия – то есть модели информационной системы. На данном уровне может быть представлена не только онтологическая модель, а так же и другие модели предметной области, например модели бизнес-процессов на языках организационного моделирования.

Заключение

Референтные модели архитектуры предприятия созданы на основе теоретических и эмпирических положений менеджмента и проектирования информационных систем и являются основой построения собственных моделей архитектуры предприятия.

Разрабатываемые таким образом онтологии могут применяться для целей анализа и синтеза архитектурных проектных решений на разных концептуальных уровнях представления предприятия. С их помощью осуществляется навигация по многоуровневым иерархическим структурам типа страт или эшелонов рамочной схемы ЕА и работа во взаимосвязанных измерениях обобщения и агрегации. Комплекс моделей поддерживает совместное использование их менеджерами и ИТ-специалистами, обеспечивая коммуникации между различными участниками ИТ-проекта: пользователями, разработчиками, программными агентами, осуществляя при этом повторное использование знаний предметной области при создании прототипов ИС в условиях трансформирующейся организации, в управлении развитием организации. ■

Литература

1. Зиндер Е., Архитектура предприятия в контексте бизнес-реинжиниринга // «Intelligent Enterprise/Корпоративные системы». – 2008. – №4 (180). – с. 36-38.
2. Эталонные модели организации деятельности в государственном секторе, Государственный университет – Высшая школа экономики, – М., ГУ ВШЭ, 2006 г.
3. Тельнов Ю.Ф. Проектирование систем управления знаниями. // Новости искусственного интеллекта. – 2002. – №4, – с. 29-34.
4. What is protégé? URL: http://protege.stanford.edu (дата обращения 30.12.2010)
5. Григорьев Л., Менеджмент по нотам, М. : Альпина, 2010. 696 с.

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ ДЛЯ УПРАВЛЕНИЯ ФИНАНСОВЫМИ РИСКАМИ

С.М. Авдошин,
профессор, руководитель отделения программной инженерии факультета бизнес-информатики Национального исследовательского университета «Высшая школа экономики»,
e-mail: savdoshin@hse.ru.

Е.Ю. Песоцкая,
кандидат экономических наук, доцент кафедры Управления разработкой программного обеспечения Национального исследовательского университета «Высшая школа экономики»,
e-mail: epesotskaya@hse.ru.
Адрес: г. Москва, ул. Кирпичная, д. 33/5.

Большинство финансовых операций проводятся в условиях неопределенности и требуют управления рисками. Система управления рисками должна включать в себя утвержденную методику и методологию управления рисками и развитую информационно-технологическую базу с использованием программного обеспечения. Выбор автоматизированных систем представляется нелегкой задачей для сотрудников организации и требует поддержки со стороны профессионалов.

Ключевые слова: информационные технологии, финансовые риски, риск-менеджмент, программное обеспечение.

Введение

Эффективная работа любой финансовой организации зависит от качества функционирования ее информационных технологий. Автоматизация процессов управления рисками значительно повышает эффективность этой работы. Операции управления финансовым риском должны выполняться в автоматическом или автоматизированном режиме с привлечением средств вычислительной техники. А уж говорить о сложной математической оценке рисков без использования современных ИТ решений просто не имеет смысла.

Существует большое число программных пакетов, поддерживающих те или иные функции управления рисками. Некоторые из них представляют собой информационные системы поддержки принятия решений, в которых присутствует модуль

управления рисками, другие являются приложениями и дополнениями различных функциональных систем, начиная с систем календарного планирования и заканчивая комплексными системами управления предприятием. Реже встречаются самостоятельные программные продукты по управлению рисками.

Процесс внедрения системы управления рисками должен осуществляться не раньше того, как банк точно определит подход к управлению и методологию оценки рисков. Это позволит минимизировать затраты на дальнейшую доработку системы и потери инвестиций в технологии, а также ускорить внедрение. При внедрении системы на слишком ранних стадиях, банк может столкнуться с необходимостью частого изменения системы из-за неполноты информации на этапах ее создания, то есть постоянных дополнительных вложений в информационные технологии.

Управление рисками как часть банковских процессов

Особенности оказываемых банком услуг таковы, что заставляют его уделять особое внимание управлению рисками. Это объясняется спецификой кредитной организации: операции, позволяющие ей получать прибыль, совершаются в основном за счет «чужих», или привлеченных средств. Одна из главных стратегических задач кредитной организации и банка, в частности, это обеспечение оптимума между доходностью и риском. Наиболее остро встает вопрос об оценке риска проводимых операций – риска неполучения дохода или риска полной потери размещенных средств, что может привести не только к потери прибыли, но и к банкротству банка.

Риски, с которыми сталкиваются сотрудники кредитных организаций, отличаются друг от друга местом, временем и причиной возникновения. Все это влияет на выбор метода управления ими. Успешное и своевременное управление рисками является важнейшим условием конкурентоспособности и надежности любого банка.

На сегодняшний день существует множество определений риска. В последнее время риск наиболее часто понимается в нескольких различных аспектах: риск как возможность (чем выше риск, тем выше потенциальный доход, но также выше и вероятные убытки), риск как опасность или угроза (негативные события, такие как финансовые потери, мошенничество, хищения, угроза репутации, ущерб или банкротство, участие в судебных процессах и т.д.), риск как неопределенность (возможность возникновения в ходе деятельности неблагоприятных ситуаций и последствий) [3]. Зачастую позитивная составляющая риска отбрасывается вовсе. В этом случае риск воспринимается банком лишь как возможная помеха успешной деятельности.

Риск-менеджмент является одной из важнейших областей современного банковского управления. Основной задачей управления рисками является предоставление формализованной информации о возможности наступления рисковых событий, выбор способов борьбы с рисковым событием и контроль реагирования [1]. Своевременное определение риска, качественная оценка и контроль позволяют значительно сократить потери и предотвратить неблагоприятные события, связанные с наступлением риска. Шаги по применению утвержденной высшим руководством стратегии по рискам документируются и полностью отражаются в системах управленческой информации банка.

Этапы управления риском

Управление риском представляет собой итерационный процесс, в котором выделяют следующие основные этапы: идентификация риска, качественный и количественный анализ риска, выбор методов воздействия на риск, контроль и корректировка результатов [5].

1. Идентификация риска – наиболее критичный этап, который заключается в составлении перечня факторов, влияющих на деятельность кредитной организации. Здесь учитываются все обстоятельства и факторы от которых зависит доходность банка. Выполнение этого шага требует глубокого знания финансовых операций, ключевых взаимосвязей производственных и управленческих процессов в банке, их связи с окружением и внешними источниками. Собранной информации должно быть достаточно для последующей обработки и анализа выявленных факторов риска.

2. Качественный анализ риска включает определение факторов риска, которые могут привести к рисковым ситуациям и возможным потерям, вероятность и время их возникновения. При этом осуществляется расчет максимальной величины потерь, если события будут развиваться по самому худшему сценарию, и рассчитана максимальная

величина потерь, которые может понести банк, с учетом управления этим же риском. Для различных рисков могут быть построены функции распределения вероятности наступления ущерба в зависимости от его размера.

3. Количественный анализ риска предполагает численное определение величин отдельных рисков и заключается в сборе и обработке статистических данных. При этом может выполняться классификация потерь с описанием их причин. Расчет всех значений рисков основывается на первичных данных (данных о сделках и остатках по счетам за выбранный статистический горизонт). Таким образом, получаемые значения риска не зависят от принятой в банке формы стандартов бухгалтерского учета и позволяют сохранить сопоставимость результатов при переходе от РСБУ к МСФО. Такой расчет особенно эффективен при наличии статистики, организованной отчетности, возможности оперативно получать данные об операциях банка и быстрой их обработки.

4. Выбор методов воздействия на риск позволяет провести сравнительную оценку эффективности для выбора оптимального решения. Сравнение методов воздействия может быть произведено на основе различных факторов – экономических, финансовых и пр. На этом этапе также определяются требуемые финансовые и трудовые ресурсы для реализации выбранного метода воздействия. Обеспечение выполнения утвержденных высшим руководством мероприятий, связанных с воздействием на риск, является важной задачей риск-менеджера.

Как правило, для каждого вида риска существует несколько основных способов воздействия.

Минимизация риска предполагает уменьшение размеров возможных потерь за счет осуществления комплекса организационно-технических мероприятий, таких как создание процедур анализа контрагентов, создание системы управления лимитами, включая распределение лимитов на отдельные сегменты бизнеса и отдельные структурные подразделения.

Принятие риска возникает вследствие невозможности или необоснованности применения мероприятий по уменьшению риска до его возникновения и предполагает отказ от любых действий, направленных на избежание риска. При этом все меры воздействия осуществляются после возникновения риска или после появления его признаков.

Передача риска означает передачу ответственности за него третьим лицам при сохранении существующего уровня риска. Передача рисков третьему лицу всегда сопровождается дополнительными выплатами, связанными с передачей обязательств, полномочий и гарантий.

Хеджирование является специфическим методом управления финансовыми рисками, которое, не влияя на частоту наступления убытков в виде уменьшения стоимости портфеля активов, с помощью специальных мероприятий уменьшает их масштаб. Однако всякое хеджирование также сопряжено с риском, определяемым, в свою очередь, как надежностью контрагента, так и системными рисками.

5. Контроль и корректировка результатов является заключительным этапом всех процессов и управления рисками. Речь идет о получении данных от менеджеров банка об убытках и принятых мерах по их устранению. На этом этапе своевременно и качественно решаются проблемы управления рисками, анализируется на сколько они адекватны размеру и объему операций. Корректировка результатов проявляется в выявлении новых обстоятельств, изменяющих уровень риска. Организация мониторинга эффективности управления рисками и проверка соответствия проводимых операций принятым процедурам может проводиться внутренними структурами банка – службами безопасности, внутреннего аудита и внутреннего контроля. Общий контроль должен осуществляться независимым финансово-аналитическим подразделением, находящимся в прямом подчинении Председателя Правления.

Классификация рисков

Для успешного управления всем множеством разнообразных видов рисков необходимо их упорядочивание с помощью системы классификации. Целью использования классификаторов является более удобное использование информации о различных видах рисков, которые отличаются друг от друга местом, временем, причиной возникновения. Это дает возможность выбирать предусмотренные методы управления рисками в зависимости от их классификации.

Существует огромное количество рисков, с которыми банк сталкивается в процессе своей деятельности. В качестве критерия разделения рисков используются области возникновения риска и источники. В соответствии с этим принципом риски

могут подразделяться на финансовые, функциональные и внешние.

Согласно представленной классификации, среди **финансовых рисков,** связанных с непредвиденными изменениями в объемах, доходности, стоимости и структуре активов и пассивов банка наиболее распространены следующие.

Процентный риск – риск потери в стоимости процентного портфеля, возникающий вследствие изменения процентных ставок по различным финансовым инструментам. Для анализа процентного риска, принимаемого банком, рассчитывается размер капитала, необходимого для покрытия процентного риска, затем оценивается его стабильность при изменении процентных ставок. В расчет процентного риска банка включаются все инструменты, имеющие переменные процентные ставки.

Современные ИТ-решения позволяют смоделировать финансовую деятельность банка методами математического программирования. Математическая модель состоит, как правило, из целевой функции, соответствующей какому-либо параметру, значение которого требуется максимизировать или минимизировать (например, максимизировать чистый процентный доход банка, минимизировать риск процентной ставки) и ее ограничений, выраженных в математическом виде.

Кредитный риск – риск уменьшения стоимости активов Банка, в особенности кредитов в результате невыполнения обязательств заемщиками. Для анализа кредитного риска проводится анализ финансового состояния эмитентов и контрагентов, обеспечение кредита, установка лимитов на операции, резервирование.

Этот вид риска возникает при осуществлении операций кредитования, формировании портфеля ценных бумаг, межбанковских и валютных операциях, работе с производными ценными бумагами и дилерской деятельности.

Существующие ИТ-решения включают как оценку и управление рисками при проведении операций с контрагентами на финансовых и фондовых рынках, так и управление рисками при коммерческом кредитовании. При оценке кредитоспособности контрагентов применяется система внутреннего кредитного рейтинга для управления рисками контрагентов. В рамках кредитной деятельности банка потенциальные заемщики подвергаются кредитному анализу с использованием балльной системы, учитывающей не только риск заемщика, но и риск кредитного продукта, и ликвидность обеспечения.

Риск ликвидности – риск недостаточности средств для соблюдения банком своих обязательств. Для анализа риска ликвидности банка определяются ключевые факторы, влияющие на ее состояние. Прогнозируются возможные изменения этих факторов. В зависимости от прогнозируемых изменений определяются сценарии поведения ликвидности банка. При анализе риска ликвидности банка рассчитывается запас ликвидных активов (моментальная ликвидность) и концентрация депозитной базы. Применение информационных технологий позволит в режиме on-line отслеживать структурные колебания активов и пассивов, разрывы по срокам размещения и привлечения капитала. Распространенный метод анализа сценариев контролирует чувствительность активов и пассивов к изменению процентных ставок и будущую потребность банка в финансовых средствах в соответствии с различными изменениями.

Ценовой риск – риск потерь вследствие изменения котировок по инструментам портфеля. Для анализа ценового риска банка включаются все ценные бумаги, по которым банк ведет операции (еврооблигации, акции, векселя и пр). Для расчета ценового риска необходимы данные по сделкам и остаткам по счетам, текущим рыночным курсам, рыночным индексам. Количественной оценкой ценового риска является максимальный размер потерь в стоимости портфеля, с принятым банком доверительном уровнем, в результате изменения рыночных цен за период прогнозирования.

В качестве ИТ-решения управления ценовым риском можно разработать систему ведения позиций по различным финансовым инструментам, позволяющую ежедневно переоценивать портфели ценных бумаг в соответствии с текущими рыночными ценами, а также системы обработки данных, выполнения расчетов, контроля соблюдения существующих требований и подготовки отчетности.

Валютный риск – это риск потерь, обусловленный неблагоприятным изменением курсов иностранных валют в ходе осуществления сделок по их купле-продаже. Он возникает только при наличии открытой позиции.

Для минимизации денежных потерь в результате колебаний валютных курсов целесообразно приме-

нять информационные системы, которые позволят в режиме реального времени отображать движение курсов валют и последнюю информацию по изменениям на мировых валютных рынках.

Рыночный риск – риск возможных потерь или неполучения планируемой прибыли вследствие неблагоприятного изменения цен на финансовых рынках. Изменение цен полностью зависит от таких значений параметров рынка, как процентные ставки, курсы валют, цены акций или товаров, корреляция между различными параметрами рынка. Причина возникновения рыночного риска заключается в неопределенности развития ситуации на финансовых рынках.

В качестве основного инструмента анализа рыночного риска используются два наиболее распространенных метода: метод расчета величины риска Value-at-Risk (VaR) методом Монте-Карло и метод анализа чувствительности портфеля к изменениям параметров рынка (Stress or Sensitivity Testing).

Инвестиционный риск – риск, связанный с инвестиционной деятельностью банка. Степень инвестиционного риска может быть уменьшена за счет снижения инфляции, принятия четкого экономического законодательства.

Распространенным средством анализа инвестиционного риска является имитационное моделирование инвестиционных процессов, которое требует использования соответствующего программного обеспечения. Метод имитационного моделирования дает наиболее точные и обоснованные оценки вероятностей при наименьших трудозатратах по сравнению с прочими методами, однако точность оценок в значительной степени зависит от качества исходных предположений и учета взаимосвязей переменных внешней среды.

К более простым методам анализа инвестиционных рисков можно отнести метод сценариев и «построение деревьев решений». Преимуществом методов является наглядность результатов и процесса анализа, недостатком – его техническая сложность реализации при большом объеме данных.

Функциональные риски, возникающие вследствие невозможности контролировать финансово-хозяйственный процесс, собирать и анализировать информацию, как правило, состоят из стратегических, технологических и операционных рисков [2].

В случае функциональных рисков использование специализированных программных средств затруднительно, поскольку классификация всех возможных сценариев функциональных рисков, в отличие от финансовых, представляется неразрешимой задачей.

Стратегический риск – риск, связанный с ошибками в стратегическом управлении из-за неверного формулирования целей организации, неверного ресурсного обеспечения их реализации и неверного подхода к управлению банковскими рисками в целом. Стратегический риск отражает также способность банка выбирать географические и продуктовые сегменты, предположительно прибыльные для банка в будущем, с учетом комплексного анализа будущей операционной среды.

Технологический риск связан с использованием различной банковских технологий и является причиной потерь из за расходов на устранение неполадок в работе оборудования. Поскольку в организации, как правило, уже используются связанные элементы банковской системы, возникают риски их несовместимости с внедряемыми технологиями, а также риски потери данных. Формирование четкой и упорядоченной методической и нормативной базы, регламентирующей проведение отдельных операций и использование существующих технологий может в значительной степени снизить подобные риски.

Операционный риск – риск несоответствия расходов банка на осуществление своих операций, потерь вследствие сбоев в выполнении каждодневных, рутинных банковских операций. К операционному риску относят функционирование систем, освоение новых операций сотрудниками банка, организационный риск соблюдения регламента выполнения действий сотрудниками. Для управления операционными рисками в банке должна существовать эффективная система контроля операций, включающая разные механизмы, действующие на различных уровнях организационной структуры, в том числе на уровне отделов и управлений. Механизмы контроля с одной стороны должны быть разработаны с целью надзора за соблюдением операционных правил и процедур соответствия деятельности каждого подразделения банка, с другой стороны – для получения возможности отслеживания возможных финансовых рисков операций и принятия оперативных мер.

Стоит обратить особое внимание на наличие такого риска, как *риск персонала/клиента*, связанный с несанкционированным доступом к данным и утечкой информации. Причиной риска персонала

является, как правило, мошенничество сотрудников внутри самой организации. Возможны риски, когда работники низшего и среднего звена банка предоставляют информацию о своих клиентах банкам-конкурентам. При этом клиентам, желающим положить значительные суммы денег на депозит, специально сообщаются заниженные депозитные ставки. Более серьезные риски утечки важной финансовой и бухгалтерской информации могут значительно повлиять на конкурентоспособность и репутацию банка.

В качестве ИТ решений по обеспечению защиты информации можно порекомендовать использование автоматизированных средств разграничения доступа и шифрования информации.

Для **внешних рисков**, которые непосредственно не связаны с работой банка, характерны такие риски, как риск изменения законодательства, экономической, социальной и политической среды. Все это может неблагоприятно повлиять на результаты банковской деятельности. Одним из основных методов управления внешними рисками, обусловленными негативным влиянием внешней среды, является страхование.

В качестве дополнительных средств анализа и контроля внешних рисков рекомендуется проводить постоянный мониторинг изменений внешней среды.

Программные решения для управления финансовыми и банковскими рисками на российском рынке

Функции управления рисками используются многими российскими банками и входят в состав практически всех современных российских банковских систем (RSBank 5, Diasoft 5NT и т.п.). В основном эти функции касаются контроля позиций по валюте (риски ликвидности) и используются в казначействе.

Среди специализированных инструментов управления финансовыми рисками можно, пожалуй, выделить три основных продукта, представленных западными разработчиками. Именно они стали наиболее применяемыми программными решениями на российском рынке.

SAS Risk Management (SAS) является широко признанным в мире решением в области управления рисками на уровне всего банка. SAS Risk Management имеет гибкую, открытую и расширяемую среду управления рисками, что позволяет в полной мере отразить российскую специфику и актуальные для российских финансовых учреждений задачи [6]. Система позволяет рассчитывать текущую и потенциальную подверженность кредитному риску, агрегировать взвешенные показатели по определенным пользователем уровням агрегации, например по контрагентам, рейтингам и т.д. Для оценки таких показателей, как Credit VaR, как и в случае управления рыночными рисками, можно использовать возможности моделирования методом Монте-Карло. Это дает возможность производить интегрированную оценку рыночного и кредитного риска, при этом принимается во внимание взаимосвязь всех вовлеченных в анализ факторов риска, что делает оценку риска более объективной.

EGAR Focus (EGAR Technology) – комплексное решение для банков и инвестиционных компаний, позволяющее осуществлять мониторинг позиций, управление рисками, оценку стоимости производных финансовых инструментов, проводить расчет прибыли и убытков в режиме реального времени [7].

Egar Focus представляет собой единую банковскую и брокерскую систему, которая имеет широкую область применения и может использоваться во фронт-, мидл- и бэк-офисных подразделениях банков, инвестиционных компаний и других финансовых организаций для мониторинга и управления рисками. За счет модульной архитектуры можно формировать оптимальный функциональный пакет для решения конкретных задач банка.

Kondor+ (Reuters) наиболее полно охватывает функциональную область дилинговых управлений инвестиционных банков, кроме того используется для оценки и контроля риска ликвидности и процентного риска [8]. Kondor+ представляет собой систему ведения позиций в режиме реального времени, с большим набором интеллектуальных и гибких средств управления сделками и широким спектром финансовых инструментов. Система позволяет анализировать прибыли и убытки, а также рассчитывать уровень риска по любому срезу торговой структуры банка и любой комбинации финансовых инструментов в режиме онлайн. На основании получаемых данных и отчетов принимаются оперативные и стратегические решения в области управления активами и пассивами банка. Дополнительный модуль KVaR+ может быть полезен для расчета и контро-

ля уровня рыночных рисков и позволяет в оперативном режиме контролировать возникновение риска по торговым и инвестиционным позициям банка в разрезе отдельных инструментов, портфелей инструментов, структурных подразделений и отдельных трейдеров.

Перечисленные решения, как уже упоминалось, представлены западными компаниями и представляют комплексные решения по управлению различными типами финансовых рисков. К сожалению, не все банки могут позволить себе использование подобных дорогостоящих систем. Поэтому во многих российских банках распространена практика создания собственных разработок в области управления рисками, предусматривающая оценку и управление основными видами банковских рисков: кредитным, рыночным и операционным.

Методика управления финансовыми рисками, разработанная сотрудниками подразделений банка, как правило, представляет инструмент математического расчета стандартной методики (Value-at-Risk, стрессовый анализ, моделирование, анализ сценариев и пр.) объединенный с системой изменения внешних и внутренних факторов, оценки эффективности инвестиций, передачи и возврата денежных средств и пр. Качество системы управления финансовыми рисками будет зависеть от полноты и актуальности информации, которая в нее поступает.

Организацию управления всеми типами рисков автоматизировать собственными силами очень сложно. Как правило, одна система обеспечивает управление лишь определенным риском. Например, в рамках кредитной деятельности банка эффективно создание скоринговых систем и систем кредитного анализа для оценки потенциальных заемщиков с использованием балльной системы, учитывающей не только риск заемщика, но и риск кредитного продукта, и ликвидность обеспечения.

Для управления рисками процессинга и распознавания способов и тенденций мошенничества с помощью банковских платежных карт возможно создание специализированных систем по распознованию нетипичных действий держателей карточек с тем, чтобы можно было вовремя идентифицировать место совершения операции и принять соответствующие меры. Подобные разработки, как правило, являются ноу-хау банка и не выходят за его пределы.

Рынок российских систем управления финансовыми рисками развит значительно слабее, чем западный. Как правило, системы представляют инструменты оценки рыночных рисков и портфельного анализа и разрабатываются в качестве приложений к банковским системам, предлагаемым компаниями-разработчиками.

Зачастую поставщиками программных средств по управлению финансовыми рисками являются привлеченные консалтинговые компании, осуществляющие ведение учета, анализа и управления рисками в финансовой организации.

Так, например, компания Accenture совместно с Risk Management Concepts Systems предлагает продукт, который позволяет рассматривать риски на различных уровнях детальности – от процедур и подразделений к оценке рисков предприятия в целом. Система *TSA (Time Series Aggregation)* позволяет агрегировать факторы риска и собирать информацию о фактических или прогнозируемых потерях. Используя «аналитический» подход система предоставляет среду для детализации факторов риска и учитывает уровень потерь по каждому подразделению и/или процессу, что делает возможным проведение анализа распределения уровня риска. Распределение рисков производится путем составления матрицы «размер ущерба/частоты возникновения» наступления рисковых событий.

Система *OpVar* компании PriceWaterhouseCoopers направлена на сбор и консолидацию данных по операционным потерям за предыдущие периоды, позволяет осуществлять моделирование на основе исторических данных.

Среди решений, построенных на основе интернет-технологий, стоит отметить созданную в компании Ernst & Young систему *Horizon*. Интернет-дизайн в соединении с функциональными возможностями обеспечивает удобство в использовании и поддерживает возможность быстрого запуска в банковских офисах по всему миру. Система позволяет идентифицировать и документировать широкий спектр рисков, принятых банковскими подразделениями, и требующих более пристального внимания. Анализ возможности уменьшения рисков и дальнейшее прогнозирование тенденций возникновения неблагоприятных событий проводится в зависимости от качественной экспертной оценки, при которой риски ранжируются и помечаются красным, желтым и зеленым цветом в зависимости от критичности.

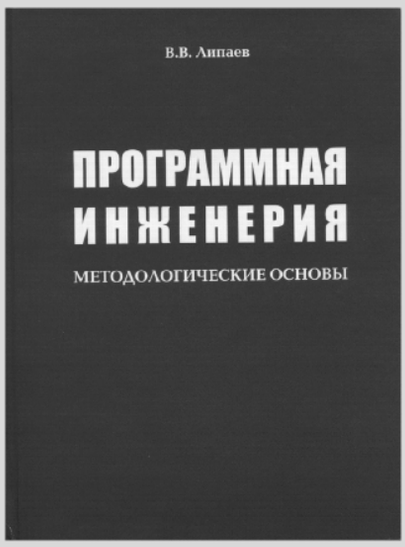
Выводы

Несмотря на широкий выбор программных средств по управлению финансовыми рисками, выбор автоматизированных систем с учетом индивидуальных особенностей предприятия представляется нелегкой задачей для сотрудников организации и требует поддержки со стороны профессионалов. Часто руководители сталкиваются перед выбором – что лучше, продолжать инвестировать в существующую систему и дорабатывать существующий продукт, или приобрести новый?

Так или иначе, без автоматизации управления и использования современных информационных технологий, банковская деятельность может превратиться в рутинный и малоуправляемый процесс, что всегда скажется на конкурентной привлекательности банка. [1] ■

Литература

1. Литовских А.М. Финансовый менеджмент: Конспект лекций. Таганрог: Изд-во ТРТУ, 1999. 76 с.
2. Лобанов А., Чугунова А. Энциклопедия финансового риск-менеджмента. М.: «Альпина Бизнес Букс», 2009, 932 с.
3. Риск менеджмент, оценка рисков. [Электронный ресурс] – http://md-hr.ru/articles/html/article32645.html
4. Basel II: Revised international capital framework. [Электронный ресурс] – http://www.bis.org/publ/bcbsca.htm
5. Duncan. B. A Guide to the Project Management Body of Knowledge // PMBOK GUIDE.– PMI, 2004.
6. Комплексное решение по управлению рисками SAS. [Электронный ресурс] -http://www.sas.com/offices/europe/russia/software/risk_dimensions/
7. Решения EGAR. [Электронный ресурс] – http://egartech.ru/
8. Reuters Kondor+ и Systematica Radius – решение по управлению рисками для фронт- и мидл-офиса. [Электронный ресурс] – http://www.systematica.ru/articles/detail.php?ID=33



В рамках Инновационной образовательной программы вышла книга
В.В.Липаева
«Программная инженерия. Методологические основы»
(Допущено УМО по образованию в области менеджмента в качестве учебника для студентов ВУЗов, обучающихся по направлению «Бизнес-информатика»).

Владимир Васильевич Липаев – профессор кафедры управления разработкой программного обеспечения НИУ ВШЭ, главный научный сотрудник Института системного программирования РАН. Около 40 лет занимается исследованиями и разработкой программного обеспечения, методов и инструментальных средств для создания управляющих программ реального времени высокого качества. Под его руководством разработаны крупные инструментальные системы программной инженерии, широко использовавшихся в оборонной промышленности и частично эксплуатируемые до настоящего времени.

Учебник содержит курс лекций, отражающий методологические основы современной программной инженерии, обеспечивающей жизненный цикл (ЖЦ) сложных программных средств (ПС).

Учебник целесообразно использовать при обучении студентов старших курсов, аспирантов и менеджеров проектов при создании сложных комплексов программ на всём их ЖЦ (64 часа лекций и 32 часа семинарских занятий). Курс ориентирован также на заказчиков, менеджеров крупных проектов, аналитиков и ведущих специалистов, обеспечивающих этапы ЖЦ сложных ПС и систем, к которым предъявляются высокие требования по качеству функционирования и ограничены доступные ресурсы разработки.

ПРОГНОЗИРОВАНИЕ ВЕРОЯТНОСТИ БАНКРОТСТВА ПРЕДПРИЯТИЙ С УЧЕТОМ ИЗМЕНЕНИЯ ФИНАНСОВЫХ ПОКАЗАТЕЛЕЙ В ДИНАМИКЕ

Т.К. Богданова,
кандидат экономических наук, доцент, заместитель заведующего кафедрой «Бизнес-аналитика» Национального исследовательского университета «Высшая школа экономики».
e-mail: tanbog@hse.ru.

Ю.А. Алексеева,
аспирантка кафедры «Бизнес-аналитика» Национального исследовательского университета «Высшая школа экономики»,
e-mail: danilova_yuliya@mail.ru.
Адрес: г. Москва, ул. Кирпичная, д. 33/5.

Предлагается комплекс логистических регрессионных моделей прогнозирования вероятности банкротства предприятий, учитывающих динамику изменения основных финансовых показателей, характеризующих финансовое состояние предприятий. Проводится сравнительный анализ прогнозной точности предлагаемых моделей, построенных на обучающей и контрольной выборках на реальных данных по предприятиям обрабатывающего производства.

Ключевые слова: логистическая регрессионная модель, финансовое состояние предприятия, прогнозирование вероятности банкротства, обучающая выборка, контрольная выборка, динамика изменения финансовых показателей.

1. Введение

Проблема оценки финансового состояния предприятий, и в частности, предсказания их возможного банкротства была и остается актуальной, поскольку в этом заинтересованы и внутренние и внешние контрагенты предприятия. Каждый участник экономического сообщества должен быть уверен в надежности и финансовой состоятельности своих партнеров, в противном случае любой имеет возможность использовать механизм банкротства как средство возврата долга неплатежеспособными партнерами. В связи с этим

руководители предприятий, менеджеры различных уровней управления должны уметь своевременно определять неудовлетворительное финансовое состояние предприятий-контрагентов на основе результатов проведенного финансового анализа, и при необходимости воспользоваться своим правом, в судебном порядке применить процедуры банкротства к должнику.

Идентификация факторов, приводящих к банкротству, и своевременное прогнозирование банкротства могут позволить руководству предприятий, испытывающих финансовые трудности, выявить причины и своевременно принять необходимые меры по оздоровлению финансового состояния предприятия.

Эта проблематика занимает умы экономистов-теоретиков и практиков-аналитиков уже на протяжении более полувека. Существует большое количество как зарубежных, так и отечественных методик, для оценки и прогнозирования финансового состояния. Тем не менее, как показывают исследования, к сожалению, зачастую они дают противоречивые результаты. Что касается зарубежных методик, то, во-первых, они базируются на зарубежной статистике 60-80 годов прошлого столетия, а во-вторых, не учитывают специфику российских предприятий. Относительно методик, предлагаемых российскими исследователями, дело обстоит по-другому. Основной причиной противоречивости результатов является то, что в основном они базируются на статистике, собранной по предприятиям розничной торговли, т.е. косвенно учитывают специфику, присущую данной отрасли. Однако в явном виде отраслевая специфика не учитывается ни одной из известных моделей. Вместе с тем, исследования показывают, что такая специфика существует [1]. Таким образом, разработка модели для оценки и прогнозирования финансового состояния российских предприятий, является весьма актуальной.

2. Постановка проблемы

Первые попытки анализа деятельности фирм-банкротов были предприняты в 30-е годы, в наиболее законченном виде методика и техника прогнозирования банкротства представлена в работах Эдварда Альтмана и Уильяма Бивера. Исследования зарубежных ученых в области предсказания банкротства предприятий позволяют сделать вывод о том, что из множества используемых коэффициентов можно выбрать лишь несколько полезных и достаточно точно предсказывающих банкротство. Однако анализ даже небольшого количества показателей требует от аналитика достаточно высокой квалификации, т.к. зачастую результаты носят противоречивый характер. Это послужило толчком к разработке моделей, которые позволяют прогнозировать банкротство предприятий, основываясь на значении одного интегрального показателя, рассчитанного по совокупности нескольких. Наиболее известными является ряд моделей, построенных на основе дискриминантного анализа: модель Альтмана [2], Р. Тафлера и Г. Тишоу [3], Фулмера, Спрингейта [4], Лиса и др., и модели, использующие для оценки вероятности банкротства логистический анализ, это модели Ольсона, Чессера и Гилбрейта. Но всех их объединяет одна общая черта. Предсказание банкротства предприятия делается на данных одного периода времени. Т.е. ни в одной из известных моделей не используются данные за ряд временных промежутков, предшествующих наступлению факта банкротства.

Тем не менее, с точки зрения здравого смысла представляется целесообразным предположить, что банкротство предприятия не является неожиданным, за исключением, наверное, тех случаев, когда оно является преднамеренным. Т.е. факту банкротства предприятия должны предшествовать события, которые могут сигнализировать об ухудшении финансового состояния предприятия, и это должно найти свое отражение в официальной финансовой отчетности и в динамике изменения финансовых показателей.

Таким образом, проблема состоит в том, чтобы выявить те финансовые показатели деятельности предприятия, изменение которых в динамике будет свидетельствовать о том, что финансовое состояние предприятия становится неудовлетворительным, построить модель, учитывающую динамику изменения финансовых показателей за несколько лет до банкротства предприятия и оценить временные горизонты прогнозирования вероятности банкротства.

3. Построение статической модели прогнозирования банкротства предприятия

При построении модели прогнозирования вероятности банкротства предприятия, в качестве целевой переменной целесообразно использовать индикатор банкротства, т.е. переменную, прини-

мающую значение 1 в случае официального банкротства предприятия и 0 – в противном случае. В пользу логистической регрессии говорит возможность моделирования бинарных целевых переменных, а также ее активное применение в области прогнозирования банкротства: примером могут служить модель Чессера, а также модель Богдановой, обеспечивающая высокую прогнозную точность в предсказании банкротств авиапредприятий [5].

Для исследования были использованы финансовые данные за период с 2000 по 2009 годы по 1357 предприятиям, деятельность которых, согласно Единому классификатору видов деятельности (ЕКВД), относится к обрабатывающему производству. Из них 173 предприятия официально были признаны банкротами в период с 2008 года по 2010 год, финансовое состояние 1184 предприятий официально считается удовлетворительным.

Для построения статической модели прогнозирования банкротства предприятий использовалась обучающая выборка, состоящая из 333 предприятий, при этом 117 предприятий в 2009 году были официально признаны банкротами, а 216 предприятий имели удовлетворительное финансовое состояние. Эти 216 предприятий, отобранные для включения в обучающую выборку из числа 1184 предприятий, официально не являющихся банкротами, были признаны имеющими удовлетворительное финансовое состояние как минимум, по 6 из 7 основных методик прогнозирования вероятности банкротства.

Оценка проводилась по следующим семи методикам: пятифакторной модели Альтмана, модели Таффлера и Тишоу, модели Фулмера для классификации банкротств, модели Чессера надзора над судами, методики Лиса, методики Спрингейта, методики Иркутской государственной экономической академии, методики Р.С. Сайфулина и Г.Г. Кадыкова. В *табл. 1* приводятся результаты оценки финансового состояния 1357 предприятий по вышеназванным 7 методикам. Как видно из табл.1, 94 предприятия или 7,9% от общего числа официально считаются имеющими удовлетворительное финансовое состояние, а, тем не менее, ни одна из вышеназванных методик таковым его не признает. С другой стороны, из числа предприятий официально признанных банкротами в 2008 году 80% были признаны имеющими удовлетворительное финансовое состояние от 1 до 5 методик.

Для построения статической модели прогнозирования вероятности банкротства предприятия была выбрана логистическая регрессия.

Уравнение прогнозной вероятности банкротства предприятия имеет следующий вид:

$$P_j = \frac{1}{1+e^{-(32{,}633-1{,}082\cdot X_{1,j}-6{,}932\cdot X_{2,j}+3{,}697\cdot X_{3,j}-5{,}712\cdot X_{4,j}-1{,}573\cdot X_{5,j})}}$$

Таблица 1.

Оценка финансового состояния предприятий по 7 основным методикам

Официальная характеристика предприятий	Предприятие не признано удовлетворительным ни по одной из 7 методик	Предприятие признано удовлетворительным по 1-2 методикам	Предприятие признано удовлетворительным по 3-5 методикам	Предприятие признано удовлетворительным по 6-7 методикам	Всего
Имеет удовлетворительное финансовое состояние	94	348	526	216	1184
	7,9%	29,4%	44,4%	18,2%	100,0%
Стало банкротом в 2008 году	7	22	6	0	35
	20,0%	62,9%	17,1%	,0%	100,0%
Стало банкротом в 2009 году	13	90	14	0	117
	11,1%	76,9%	12,0%	,0%	100,0%

где P_j – вероятность банкротства предприятия *j*;

$X_{1,j}$ – общая оборачиваемость активов предприятия *j*;

$X_{2,j}$ – отношение чистой прибыли к активам предприятия *j*;

$X_{3,j}$ – отношение краткосрочных и долгосрочных заемных средств к активам предприятия *j*;

$X_{4,j}$ – отношение долгосрочных обязательств к активам предприятия *j*;

$X_{5,j}$ – натуральный логарифм выручки предприятия *j*.

Данная модель была названа статической, т.к. она не учитывает динамику изменения финансовых показателей в течение ряда лет.

Результаты тестирования точности статической модели на обучающей выборке предприятия показали, что модель правильно спрогнозировала 202 из 216 предприятий (94%), имеющих удовлетворительное финансовое состояние и 102 из 117 предприятий (87%), которые были официально признаны банкротами в 2009 году. Всего были правильно классифицированы 91% предприятий.

Разработанная статическая модель показала достаточно высокую прогнозную точность при тестировании на контрольной выборке. В контрольную выборку вошло 381 предприятие[1] с удовлетворительным финансовым положением и 56 предприятий, ставших банкротами в 2008 и 2010 годах. Даже за 3 года до расчетного периода модель правильно спрогнозировала свыше 73 % предприятий, ставших банкротами, причем точность прогноза с каждым годом увеличивалась вплоть до 91% в год банкротства. Даже за 5 лет до начала процедуры банкротства модель правильно идентифицировала свыше 50% предприятий-банкротов. В отношении предприятий с удовлетворительным финансовым состоянием модель правильно классифицировала 296 из 381 предприятия. Таким образом, точность классификации предприятий составила 78%.

Улучшение качества логистической регрессии при включении переменных было подтверждено с помощью одного из индикаторов качества подгонки – логарифма правдоподобия. Для построенной модели с пятью переменными-регрессорами начальное значение -2LL=349,346 было уменьшено на 248,725, и стало равным 100,621, что свидетельствует о значительном улучшении качества модели. Также качество построенной модели было подтверждено псевдо R2 Нагелькерка, высокой на уровне 1% значимостью критерия хи-квадрат и критерием Хосмера и Лемешова.

4. Выявление основных сценариев развития банкротства предприятий

Была выдвинута следующая гипотеза. Процедуре банкротства предприятий могут предшествовать разные сценарии развития событий. Эти сценарии, так или иначе, найдут свое отражение в динамике изменения финансовых показателей, вошедших в регрессионную статическую модель. Если при построении модели, прогнозирующей вероятность банкротства предприятия, учесть сценарии развития событий, то такая модель может дать существенно более высокую точность прогнозирования.

Для выявления возможных сценариев изменения финансовых показателей предприятия в динамике, закончившихся его банкротством, все предприятия, обанкротившиеся в 2009 году, с использованием метода иерархической кластеризации были разделены на группы, и для каждой из групп предприятий был проведен анализ характера изменения финансовых показателей за несколько лет до банкротства.

В качестве переменных для разбиения предприятий на кластеры были взяты вероятности банкротства предприятий за период с 2004 по 2009 годы, рассчитанные для каждого предприятия-банкрота по полученной статической модели (1).

Для анализа использовалось 3-х кластерное решение. В первый кластер вошли 35 предприятий, во второй – 46, в третий – 36 предприятий. По каждому из трех кластеров была рассчитана средняя вероятность банкротства предприятий. На рис. 1 показано изменение средних значений вероятности банкротства в период с 2000 по 2009 годы для каждой кластерной группы.

Как видно из *рис. 1*, к кластеру №1 относятся предприятия, у которых вероятность банкротства на протяжении первых пяти лет с 2000 г. до 2004 г. была невысокая, более того, даже уменьшалась вплоть до 2004 года, затем она резко выросла в 2005 году и ежегодно увеличивалась, пока ситуация не завершилась банкротством предприятия в 2009 году.

[1] Из 1184 предприятий с удовлетворительным финансовым положением 216 предприятий вошли в выборку для построения модели. Из оставшихся 968 предприятий было выбрано 381 предприятие, финансовое состояние которых было оценено как устойчивое по 4 или 5 из 7 основных методик.

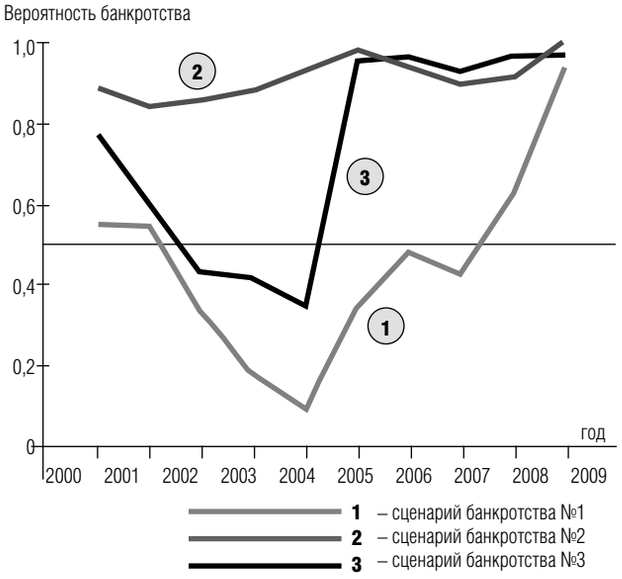


Рис. 1. График изменения средней вероятности банкротства предприятий для каждой кластерной группы в динамике.

В кластер №2 попали предприятия, у которых финансовое состояние оставалось неудовлетворительным на протяжении всех десяти лет. Т.е. прогнозная вероятность банкротства была выше 0,8. Тем не менее, банкротство этих предприятий фактически наступило только в 2009 году.

К кластеру №3 отнесены предприятия, вероятность банкротства которых в 2000 и 2001 годах являлась достаточно высокой, хотя явно прослеживалась тенденция к уменьшению вероятности банкротства. Начиная с 2002 по 2004 годы, наблюдалась тенденция к улучшению финансового состояния, что нашло отражение в прогнозном значении вероятности банкротства. Однако в 2005 году прогнозная вероятность банкротства резко выросла и оставалась практически постоянной на протяжении нескольких лет вплоть до банкротства в 2009 году.

На *рис. 2* хорошо видны кластеры, на которые распалась анализируемая выборка из 117 предприятий-банкротов. Т.е. подтверждается выдвинутая гипотеза о том, что изменение финансового состояния предприятия в течение ряда лет носит не хаотический характер, а развивается по вполне определенному сценарию. Отсюда было выдвинуто предположение, что можно улучшить статическую модель прогнозирования банкротства, если учесть изменение финансовых показателей за несколько лет до наступления банкротства, т.е. учесть, по какому из трех выявленных сценариев, развивается ситуация, способная с течением времени привести каждое конкретное предприятие к возможному банкротству.

5. Построение комплекса динамических моделей

Для построения комплекса динамических моделей использовались следующие переменные:

✦ вероятность банкротства предприятия в 2005 году;

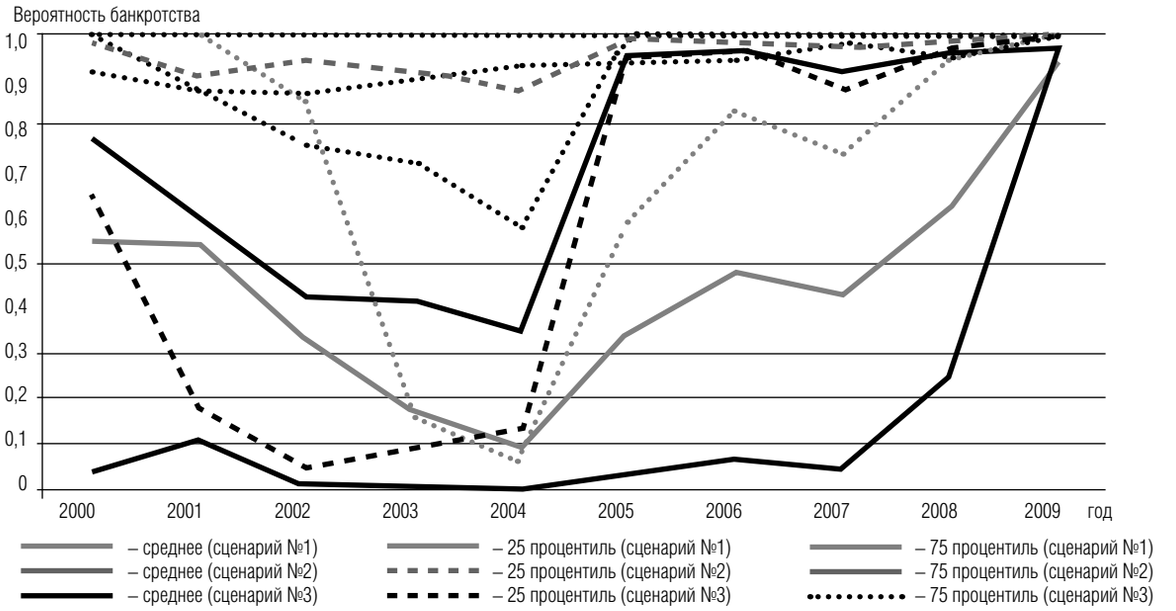


Рис. 2. График изменения средней вероятности банкротства предприятий для каждой кластерной группы в динамике.

✦ вероятность банкротства предприятия в 2006 году;

✦ вероятность банкротства предприятия в 2007 году;

✦ вероятность банкротства предприятия в 2008 году;

✦ отношение вероятности банкротства предприятия в 2005 году по сравнению с 2004 годом;

✦ отношение вероятности банкротства предприятия в 2006 году по сравнению с 2005 годом;

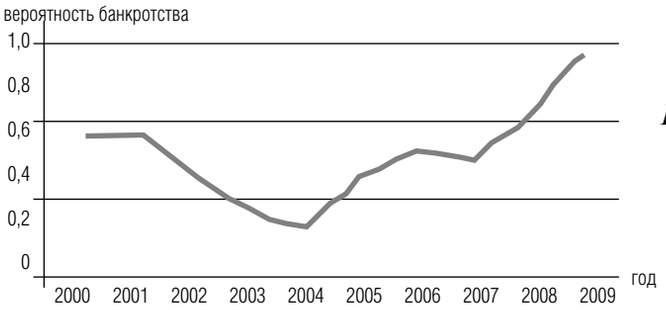
✦ отношение вероятности банкротства предприятия в 2007 году по сравнению с 2006 годом;

✦ отношение вероятности банкротства предприятия в 2008 году по сравнению с 2007 годом

В результате в соответствии с каждым из трех сценариев была построена логистическая модель прогнозирования вероятности банкротства предприятия, учитывающая сценарий развития событий на протяжении 10 лет, приводящий предприятие к банкротству в 2009 году. Т.е. косвенно в полученных моделях было учтено изменение финансовых показателей в динамике за десятилетний период.

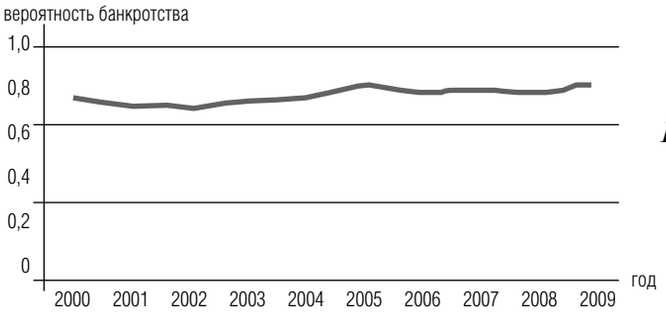
В приведенных ниже моделях используются следующие обозначения:

$P_{j,t}$ – вероятность банкротства предприятия j в год t.



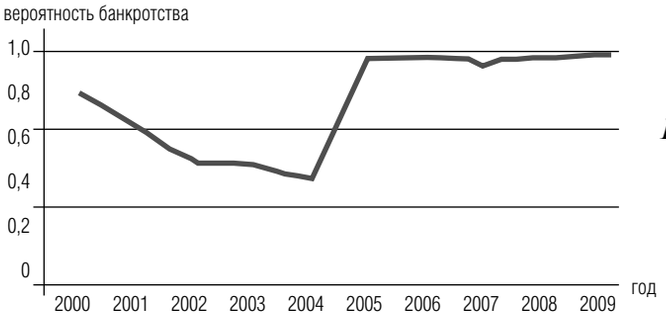
$$P_{jt+1} = \frac{1}{1+e^{-(9{,}912\cdot P_{jt+1}+0{,}213\cdot\frac{P_{jt}}{P_{jt-1}}X_{1,j}-3{,}58)}}$$

Рис. 3. Динамическая модель №1 прогнозирования вероятности банкротства предприятий с учетом того, что изменение финансового состояния происходит по сценарию №1.



$$P_{jt+1} = \frac{1}{1+e^{-(6{,}782\cdot P_{jt}+4{,}803\cdot P_{jt-2}-6{,}211)}}$$

Рис. 4. Динамическая модель №2 прогнозирования вероятности банкротства предприятий с учетом того, что изменение финансового состояния происходит по сценарию №2.



$$P_{jt+1} = \frac{1}{1+e^{-(12{,}944\cdot P_{jt}-8{,}412)}}$$

Рис. 5. Динамическая модель №3 прогнозирования вероятности банкротства предприятий с учетом того, что изменение финансового состояния происходит по сценарию №3.

6. Результаты тестирования динамических моделей

Оценка качества комплекса динамических моделей прогнозирования вероятности банкротства проводилась на обучающей и контрольной выборках.

а. Оценка качества динамических моделей на обучающей выборке

Проверка точности динамических моделей на выборке предприятий, банкротство которых развивалось по сценарию №1

В выборку для тестирования вошли 35 предприятий, ставших банкротами в 2009 году и попавших в результате процедуры кластеризации в кластер №1, и 104 предприятия, финансовое состояние которых было признано удовлетворительным по семи методикам.

В год банкротства все три модели с высокой точностью идентифицировали предприятий – банкротов. Однако при увеличении количества лет до банкротства точность модели №1 оказалась значительно выше точности моделей №2 и №3.

За 4 года до банкротства динамическая модель №1 корректно прогнозировала высокую вероятность банкротства у 77% предприятий, в то время как точность модели №2 и модели №3 оказалась равной 26% и 11% соответственно.

Проверка точности динамических моделей на выборке предприятий, банкротство которых развивалось по сценарию №2

В выборку для тестирования вошли 46 предприятий, ставших банкротами в 2009 году и попавших в результате процедуры кластеризации в группу №2, и 104 предприятия, финансовое состояние которых было признано удовлетворительным по семи методикам.

Как видно из *рис. 7*, все три динамические модели с высокой точностью идентифицировали предприятий – банкротов на каждом временном промежутке тестирования моделей. У анализируемых предприятий в течение многих лет средние значения финансовых показателей значимо отличались от средних значений показателей успешных предприятий отрасли. Это позволило спрогнозировать высокую вероятность банкротства для таких предприятий со 100% точностью даже за 4 года до банкротства.

Соотношение сценария изменения вероятности банкротства и точности динамических моделей №1, №2, №3.

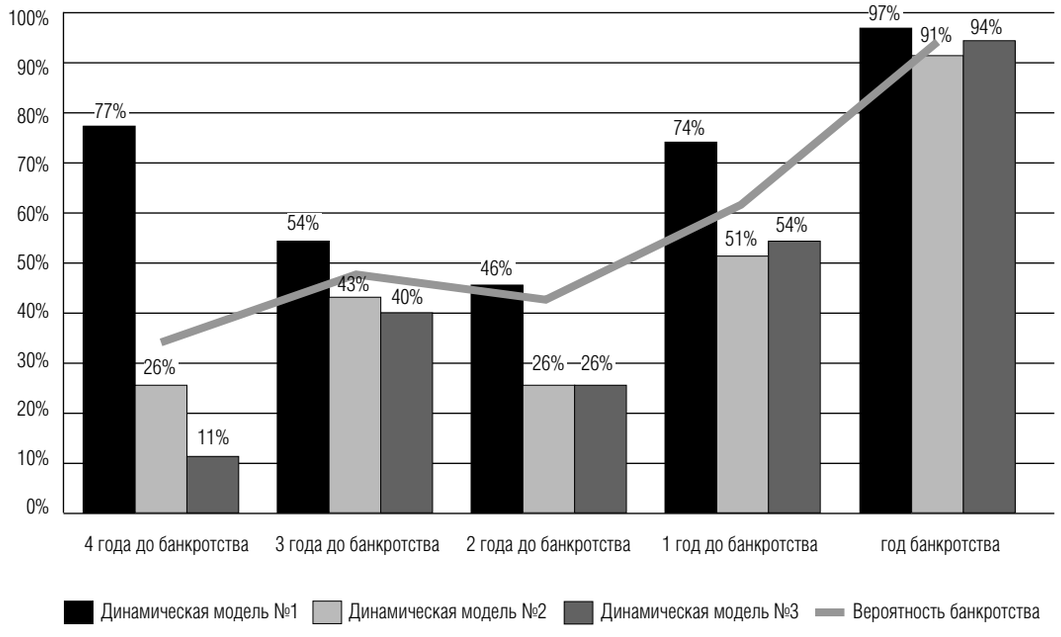


Рис. 6. Проверка точности динамических моделей №1, №2, №3 на выборке предприятий, банкротство которых развивалось по сценарию №1.

Рис. 7. Проверка точности динамических моделей №1, №2, №3 на выборке предприятий, банкротство которых развивалось по сценарию №2.

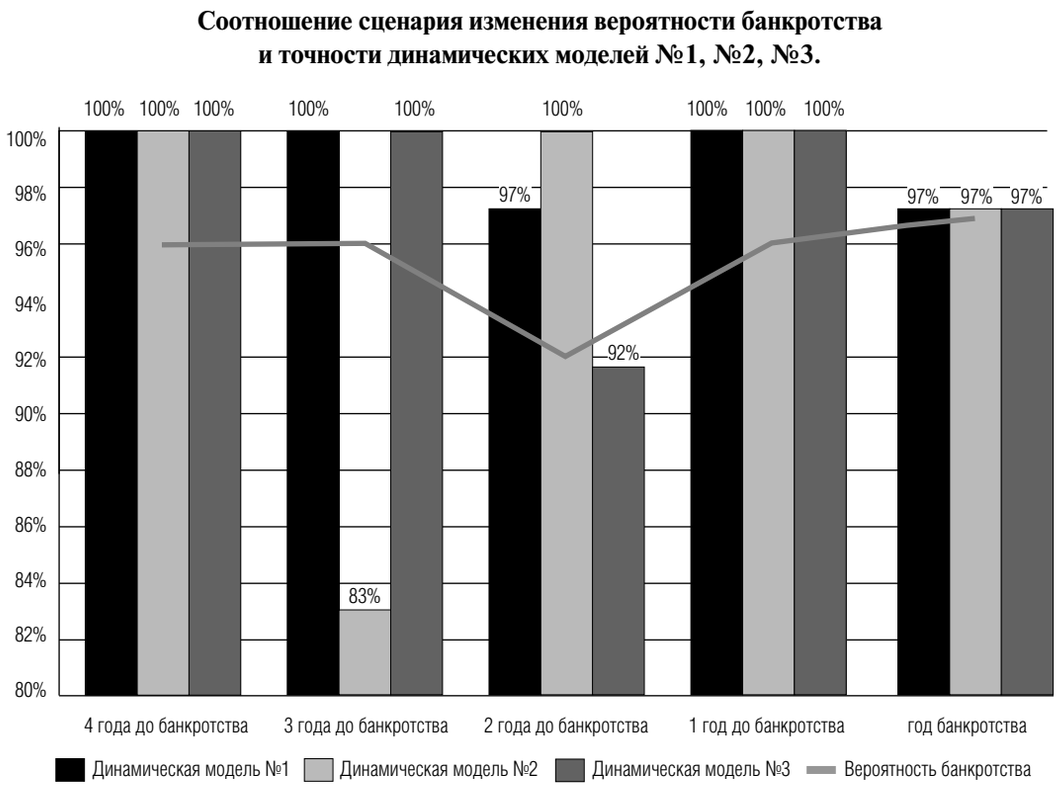


Рис. 8. Проверка точности динамических моделей №1, №2, №3 на выборке предприятий, банкротство которых развивалось по сценарию №3.

Проверка точности динамических моделей на выборке предприятий, банкротство которых развивалось по сценарию № 3

В выборку для тестирования вошли 36 предприятий, ставших банкротами в 2009 году и попавших в результате процедуры кластеризации в группу №3, и 104 предприятия, финансовое состояние которых было признано удовлетворительным по семи методикам.

Как видно из *рис. 8*, результаты тестирования динамических моделей в данном случае аналогичны результатам тестирования моделей на финансовых данных предприятий из кластера №2. Это объясняется тем, что, начиная с 2005 года, финансовое положение предприятий кластера №3 резко ухудшилось, и стало сопоставимо с финансовым состоянием предприятий из кластера №2.

Сравнение точности динамических моделей при идентификации предприятий с удовлетворительным финансовым состоянием

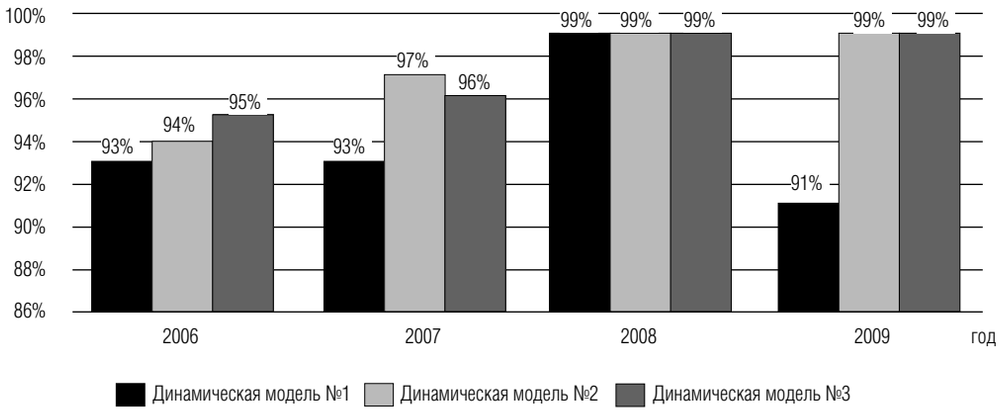


Рис. 9. Проверка точности динамических моделей №1, №2, №3 при идентификации предприятий с удовлетворительным финансовым состоянием.

Проверка точности динамических моделей на выборке предприятий, имеющих удовлетворительное финансовое положение

В обучающую выборку для проверки точности классификации предприятий с удовлетворительным финансовым состоянием вошли 104 предприятия, финансовое состояние которых было признанно устойчивым по семи методикам оценки. Как видно из *рис. 9*, все три динамические модели правильно классифицируют предприятия, признанных имеющими удовлетворительное финансовое состояние. За 4-х летний период точность классификации не менее 91%. Так в 2009 году динамическая модель №1 корректно идентифицировала 95 из 104 предприятий (91%), динамические модели №2 и №3 правильно классифицировали 103 из 104 предприятий (99%).

Помимо высокой точности классификации предприятий, качество всех трех динамических моделей было подтверждено псевдо R2 Нагелькерка, высокой значимостью критерия хи-квадрат и критерием Хосмера и Лемешова

b. Оценка качества динамических моделей на контрольной выборке

Далее прогнозная точность комплекса динамических моделей была проверена на данных, которые не использовались при построении модели. В контрольную выборку вошли 514 предприятий, из них: 21 предприятие, ставшее банкротом в 2010 году, и 493 предприятия, финансовое состояние которых было признано удовлетворительным по четырем, пяти или шести методикам оценки финансового состояния.

На *рис.10* приведен сравнительный анализ точности прогнозирования вероятности банкротства предприятий с использованием динамических моделей в различные периоды времени до официального признания банкротства в 2010 году.

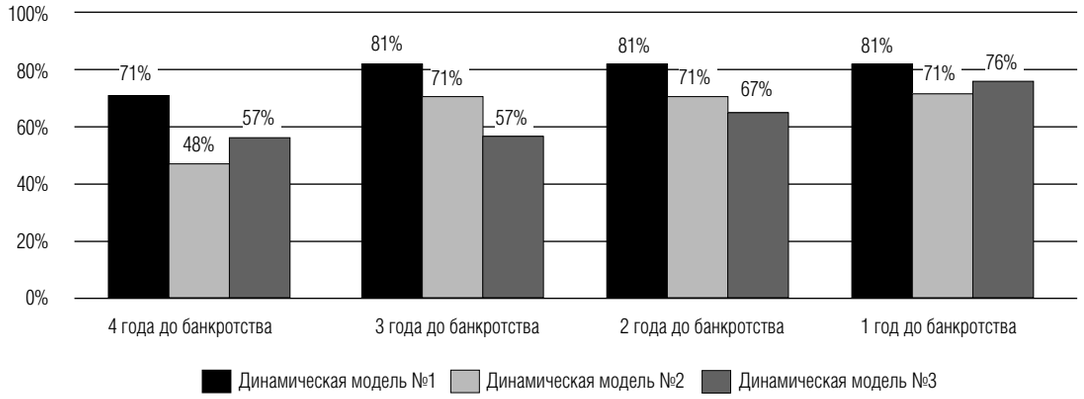


Рис. 10. Динамика изменения точности моделей при идентификации предприятий-банкротов за разное количество лет до банкротства.

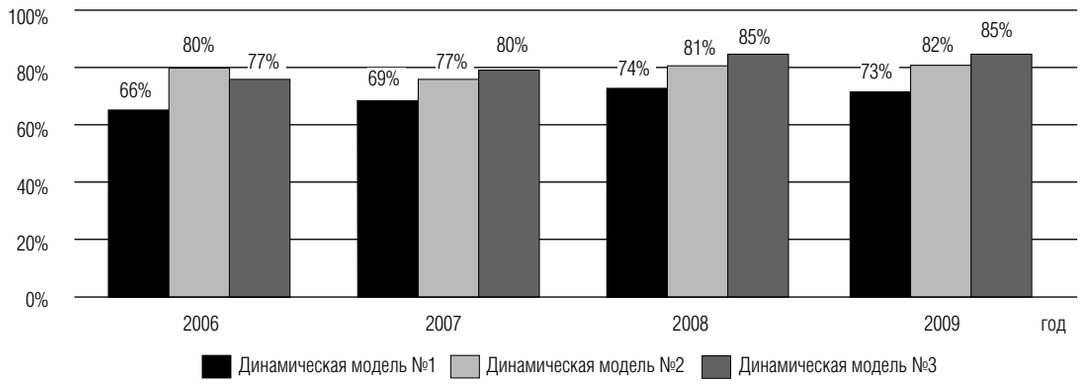


Рис. 11. Изменение точности динамических моделей при идентификации предприятий с удовлетворительным финансовым состоянием за период с 2006 по 2009 год.

Из трех моделей наиболее точной оказалась модель №1, учитывающая темп изменения вероятности банкротства во времени. При увеличении количества лет до банкротства точность всех трех моделей снижалась.

В отношении предприятий с удовлетворительным финансовым положением по данным 2009 года динамическая модель №1 правильно классифицировала 361 из 493 (73%) предприятий, динамическая модель №2 – 405 из 493 (82%) предприятий, динамическая модель №3 – 419 из 493 (85%) предприятий.

Таким образом, можно сделать вывод, что разработанный комплекс динамических моделей, учитывающих изменение финансовых показателей во времени, обладает высокой прогнозной точностью на длительном временном промежутке.

Заключение

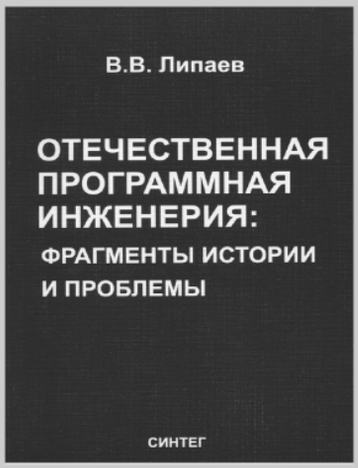
В данной статье был предложен методологический подход для прогнозирования вероятности банкротства предприятий обрабатывающего производства, учитывающий динамику изменения финансовых показателей. Этот подход может быть использован для прогнозирования вероятности банкротства производственных предприятий и других отраслей.

В зависимости от сценария изменения вероятности банкротства можно прогнозировать вероятность банкротства предприятия за период от года до четырех лет до банкротства с достаточно высокой степенью точности. Для этого был авторами разработан и апробирован на выборке предприятий обрабатывающего производства ком-

плекс моделей оценки финансового состояния и прогнозирования вероятности банкротства предприятия, учитывающих динамику изменения финансовых показателей и обладающих высокой прогнозной точностью на длительном временном промежутке. ■

Литература

1. Богданова Т.К. Применение методов многомерного статистического анализа для прогнозирования вероятности банкротства российских предприятий // Труды IX Международной конференции «Применение многомерного статистического анализа в экономике и оценке качества» (24-26 августа 2010 г.). – ГУ-ВШЭ, 2010. – с. 64–67.
2. Ковалев В.В. Финансовый анализ. – М.: Финансы и статистика, 2000. – 512 с.
3. Зевайкина С.Н. Диагностика вероятности банкротства организации. – М.: Финансы и статистика, 2002. - 412 с.
4. Арутюнян А.Б. Опыт применения моделей Фулмера и Спрингейта в оценке венгерских предприятий сельского хозяйства // Аудит и финансовый анализ, 2002, №2. – с. 200–204.
5. Богданова Т.К., Баклакова А.В. Инструментальные средства прогнозирования вероятности банкротства авиапредприятий // Бизнес-информатика, 2008, №1. – с. 45–61.
6. Данилова Ю.А. Проблемы прогнозирования банкротства // Проблемы теории и практики управления, 2009, №9. – с. 65–70.
7. Богданова Т.К., Данилова Ю.А. Моделирование финансового состояния компаний // Сборник научных трудов международной научно-практической конференции «Перспективные инновации в науке, образовании, производстве и транспорте», 2010, Т.19. – с. 82–87.
8. Бююль А., Цефель П. SPSS: искусство обработки информации. Анализ статистических данных и восстановление скрытых закономерностей. – М.: Диасофт, 2005. – 602 с.
9. Яковлева И.Н. Как спрогнозировать риск банкротства компании в системе риск- менеджмента // Справочник экономиста, 2008, №4.



Издательство «Синтег» выпустило новую книгу
Владимира Васильевича Липаева,
профессора кафедры управления программной инженерии НИУ-ВШЭ и главного научного сотрудника Института системного программирования РАН
«Отечественная программная инженерия: фрагменты истории и проблемы».

В монографии проанализированы этапы отечественной истории развития вычислительной техники с акцентом на методы и процессы программирования. Первая глава отражает развитие в стране автоматизации программирования в 50–60-е гг. Представлены процессы, начальные проекты отечественной вычислительной техники, развитие программирования и роль ведущих специалистов, заложивших основы в этой области. Выделены особенности развития специализированных вычислительных машин и программирования для оборонных систем реального времени. Формированию программной инженерии в 70-е гг. посвящена вторая глава. В третьей главе отражено развитие программной инженерии в 80-е гг. Изложена история развития экономики, методов и процессов программной инженерии в 70–80-е гг. Значительное внимание уделено реализации ПРОМЕТЕЙ-технологии программной инженерии для создания крупных комплексов программ реального времени оборонных систем. В четвертой главе подведены итоги развития программной инженерии и формирования ее методологии. Представлены проблемы расширения состава и совершенствования международных стандартов и инструментария программной инженерии, а также проблемы обучения методологией программной инженерии студентов и специалистов.

Книга предназначена для специалистов по вычислительной технике и программной инженерии, студентов и аспирантов, интересующихся историей развития и проблемами отечественной науки и техники в этой области.

О НОВОМ ПОДХОДЕ К СЕМАНТИЧЕСКОМУ ПРЕОБРАЗОВАНИЮ ЕСТЕСТВЕННО-ЯЗЫКОВЫХ ЗАПРОСОВ ПОИСКОВЫХ СИСТЕМ

А.В. Кириллов,
аспирант кафедры Инноваций и бизнеса в сфере информационных технологий факультета бизнес-информатики Национального исследовательского университета «Высшая школа экономики»,
e-mail: antonv.kirillov@gmail.com.

В.А. Фомичев,
доктор технических наук, профессор кафедры Инноваций и бизнеса в сфере информационных технологий факультета бизнес-информатики Национального исследовательского университета «Высшая школа экономики»,
e-mail: vfomichov@hse.ru.
Адрес: г. Москва, ул. Кирпичная, д. 33/5.

В статье рассматриваются существующие подходы к поиску информации, анализируются их основные недостатки. Предлагается новый подход к поиску информации, основанный на семантическом преобразовании поисковых запросов. В рамках предлагаемого подхода выделяются классы естественно-языковых запросов, предлагаются формальная модель проблемно-ориентированной системы первичных единиц концептуального уровня и метод построения семантического расширения поискового запроса, а также иллюстрируется применение метода.

Ключевые слова: семантический поиск, семантический анализ, теория К-представлений, естественно-языковые запросы, поисковые системы.

Введение

Несмотря на большое разнообразие подходов к поиску информации [1, 2, 3], фундаментальной проблемой при разработке поисковых систем является определение релевантности – ситуации, когда документ соответствует запросу. Для определения релевантности документов используется большое разнообразие методов, таких как VSM (векторно-пространственная модель), функции BM25 и BM25F (учитывающие различные весовые факторы слов в документе), функции Okapi, Ponte, алгоритм LCA и другие.

Однако системы, реализующие поиск по ключевым словам, сталкиваются с серьезными проблемами, связанными с качеством результатов поиска. Часто возникает ситуация, когда результаты поиска различны для двух запросов с одинаковым смыслом, но отличающихся грамматически.

Семантическая поисковая система в ее классическом виде оперирует с мета-данными, описывающими документы. Данное описание хранится в таких форматах, как RDF, RDFS, OWL и другие. Мета-информация позволяет получить семантическое описание содержимого документов. В настоящее время семантические поисковые системы используют большое разнообразие методов и технологий, большинство которых основано на анализе естественного языка. Подходы, применяемые в последние годы в семантическом поиске, весьма разнообразны: увеличение семантической релевантности посредством дополнительного синтаксического анализа и использования обнаруженных данных RDF [3], анализ поисковых запросов и документов на основе триплетов с использованием онтологии предметной области [4], использование автоматически сгенерированных онтологий при поиске [5], вопросно-ответные системы на основе семантических графов и анализа триплетов [6], извлечение семантических отношений естественного языка с помощью шаблонов грамматических зависимостей [7] и многие другие.

Хотя сегодня имеется много различных подходов к семантическому поиску в Интернете, остаются не полностью решенными следующие задачи: разработка естественно-языковых анализаторов поисковых запросов, определение типа вопроса, определение объекта интереса поискового запроса, определение предметной области либо областей, к которым может относиться поисковый запрос, определение принадлежности найденных документов той же предметной области, что и объект интереса поискового запроса, поиск синонимов объекта интереса поискового запроса и некоторые другие. Решение данных задач достигается предлагаемым подходом к семантически-ориентированному поиску.

Описание предлагаемого подхода

Предлагаемый подход к реализации семантически-ориентированного поиска базируется на следующих положениях: в силу того, что семантическое описание документов в большинстве случаев недоступно, алгоритмизация создания данного описания является весьма трудоемкой задачей, технологии семантического веба и гипертекстового поиска развиваются параллельно и независимо друг от друга, предлагается реализовать семантически-ориентированную поисковую систему, выполняющую преобразование поискового запроса, в зависимости от его типа, в форму, позволяющую синтаксической поисковой системе найти адекватное подмножество документов, семантически соответствующих ожиданиям пользователя. Данный подход оперирует с разными типами вопросов на естественном языке и позволяет создавать семантическое описание для каждого из них. В случае, если какой-то из введенных пользователем вопросов не может быть проанализирован, пользователь получит результаты работы системы поиска по ключевым словам.

Для успешного выполнения преобразования поискового запроса к расширенному виду необходимы следующие обязательные элементы:

1. Математическая модель проблемно-ориентированной системы первичных единиц концептуального уровня, используемых поисковой системой.
2. Математическая модель многообразия смысловых структур как ориентир для построения семантических представлений естественно-языковых поисковых запросов.
3. Математическая модель лингвистической базы знаний, позволяющая связать семантическое представление поискового запроса с его грамматическим и семантическим окружением.
4. Алгоритм преобразования поискового запроса в первичное семантическое представление с последующим анализом и расширением.
5. База знаний, отражающая взаимосвязи семантических единиц, а также включающая механизм определения отношений между ними.

Анализ научной литературы показал, что теория К-представлений, предложенная В.А. Фомичевым [1, 8], по ряду характеристик наиболее удобна для решения поставленной задачи создания семантического представления поискового запроса.

Типизация запросов на естественном языке

По результатам проведенных исследований были выявлены, проанализированы и сгруппированы вопросы на естественном языке, а также вы-

делены следующие группы вопросов и методы их обработки:

✦ общие (традиционные вопросы, не касающиеся специфики того или иного объекта интереса). В процессе преобразования запросов будут использованы синонимы, гипонимы и гиперонимы объекта интереса запроса, на этапе анализа возвращаемых документов будут использоваться антонимы объекта интереса

✦ аспектно-ориентированные (вопросы, касающиеся характеристик объекта интереса, либо его особенностей). При анализе определяется принадлежность объекта интереса к предметным областям, используются определения, синонимы, антонимы и т.д.

✦ вопросы, касающиеся сохранения или изменения состава того или иного множества; при анализе используется база знаний, содержащая некоторый набор сведений о тех или иных объектах и множествах объектов.

✦ вопросы достижения целей (вопросы, связанные с успехами и неудачами тех или иных интеллектуальных систем).

Формализация обработки перечисленных вопросов, помимо общих, ранее в доступной научной литературе не рассматривалась.

В рамках проводимого исследования была выделена группа наиболее актуальных и практических значимых вопросов для их детального анализа и разработки алгоритма семантического преобразования, а также алгоритма обработки данного вида вопросов. Таковыми являются вопросы аспектно-ориентированного типа. Введем формальное определение аспектно-ориентированных вопросов.

Аспектно-ориентированными вопросами будем называть вопросительные предложения, в которых запрашивается информация, касающаяся различных аспектов того или иного объекта или системы. Такими аспектами могут являться характеристики, условия существования или функционирования, назначение, структурная организация, функции, области применения, принадлежность к какому-либо классу, принципиальные отличия, особенности и возможности различных объектов и систем.

Для представления различных аспектов необходимо ввести реляционные символы, строго соответствующие тому или иному аспекту объекта или системы. Были выделены 11 основных аспектов, и для каждого из них введен специальный реляционный символ. Рассмотрим данные символы, их смысл и пример вопроса, в соответствие которому может быть поставлен каждый реляционный символ (под X и Y будем понимать объекты интереса поискового запроса, если не указано другого):

(1) *Описание_структуры* – данный символ предназначен для представления вопросов вида «Как устроен X?». Например: «Как устроен двигатель внутреннего сгорания?»;

(2) *Описание_характеристик* – данный символ предназначен для представления вопросов вида «Каковы характеристики X?». Например: «Каковы характеристики автомобиля Mercedes ML 350?»;

(3) *Описание_работы* – данный символ предназначен для представления вопросов вида «Как работает X?». Например: «Как работает аппарат магнитно-резонансной томографии?»;

(4) *Описание_функций* – данный символ предназначен для представления вопросов вида «Каковы функции X?». Например: «Каковы функции системы менеджмента качества?», «Какие функции выполняет сервер локальной сети?»;

(5) *Описание_назначения* – данный символ предназначен для представления вопросов вида «Для чего предназначен X?», «Каково назначение X?». Например: «Для чего предназначен реостат?», «Каково назначение межкомпьютерной связи?»;

(6) *Описание_применения* – данный символ предназначен для представления вопросов вида «Где используется X?», «Как применять X?». Например: «Где используется Java?», «Как применять активную XSS?»;

(7) *Описание_принадлежности* – данный символ предназначен для представления вопросов вида «К какому классу принадлежит X?», «К какой категории относится X?». Например: «К какому классу соединений относятся жиры?», «К какой категории относятся офисы?»;

(8) *Описание_различий* – данный символ предназначен для представления вопросов вида «Чем отличается X от Y?», «В чем разница между X и Y?». Например: «Чем отличается архитектура x86 от x64?», «В чем разница между процессорами Dual Core и Core 2 Duo?»;

(9) *Описание_общих_характеристик* – данный символ предназначен для представления вопросов вида «Что общего у X с Y?», «Каковы общие черты X и Y?». Например: «Что общего у резины и каучука?», «Каковы общие черты финансов и денег?»;

(10) *Описание_особенностей* – данный символ предназначен для представления вопросов вида «Как ведет себя X [в ситуации Y]?», «Каковы особенности работы X [в условиях Y]?», где X – объект

интереса поискового запроса, а Y – опциональная часть вопроса, уточняющая вопрос, служащая дополнительным условием (ограничением). Например: «Как ведет себя аргон при повышенном давлении?», «Каковы особенности работы буровой установки при высокой температуре?»;

(11) *Описание_возможностей* – данный символ предназначен для представления вопросов вида «Каковы возможности X?». Например: «Каковы возможности платформы .NET?».

Следует отметить, что перечисленные виды вопросов могут и должны дополняться новыми типами для обеспечения большей степени покрытия потребностей поисковой системы. Для поддержки построения семантического представления поисковых запросов необходима гибкая расширяемая математическая модель системы первичных единиц концептуального уровня. Построим такую модель, используя в качестве отправной точки определения сортовой системы и концептуально-объектной системы из [1, 8].

Математическая модель проблемно-ориентированной системы первичных единиц концептуального уровня

В монографиях [1, 8] вводится базовая математическая модель для описания системы первичных единиц концептуального уровня, используемых прикладной интеллектуальной системой. Эта модель определяет новый класс формальных объектов, называемых *концептуальными базисами (к.б.)*. Каждый к.б. *B* строится для формализации определенной группы предметных областей и задает формальный язык *Ls(B)*, называемый *СК-языком (стандартным концептуальным языком) в базисе B*. Язык *Ls(B)* предназначен для построения семантических представлений (СП) произвольно сложных текстов, относящихся к рассматриваемой группе областей. Произвольный к.б. *B* является упорядоченной тройкой вида *(S, Ct, Ql)*, где *S, Ct* – формальные объекты, называемые соответственно *сортовой системой и концептуально-объектной системой*, а *Ql* – формальный объект, называемый *системой кванторов и логических связок*.

Сортовой системой в [1, 8] называется произвольная упорядоченная четверка *S* вида *(St, P, Gen, Tol)*, где *St* – конечное множество символов; *P* – элемент множества *St; Gen* – непустое бинарное отношение на *St*, являющееся частичным порядком на *St* (т. е. рефлексивным, транзитивным и антисимметричным); *Tol* – бинарное отношение на *St*, являющееся антирефлексивным и симметричным, и выполняется несколько дополнительных условий.

Элементы множества *St* называются *сортами; P – сортом «смысл сообщения»*; $\mathrm{Gen} \subset St \times St$ – *отношением общности;* $Tol \subset St \times St$ – *отношением совместимости* (толерантности). Если пара *(u, t)* входит в *Gen*, то можно использовать эквивалентную запись $u \to t$ и говорить, что *t* – конкретизация сорта *u*, а *u* – обобщение сорта *t*. Если $(s, u) \in Tol$, то используется запись $s \perp u$ и говорится, что сорт *s* совместим с сортом *u*.

Отношение общности *Gen* отражает существование иерархии (по степени общности) наиболее общих понятий – сортов. Например, для некоторого к.б. *B* отношение *Gen* может включать пары *(физич. объект, динамич. физич. объект), (физич. объект, физич. объект)*. Отношение совместимости *Tol* отражает существование различных, несопоставимых («ортогональных») семантических характеристик некоторых сущностей из рассматриваемой группы предметных областей. Например, человек одновременно является интеллектуальной системой и динамическим физическим объектом. Поэтому для некоторого к.б. *B* отношение *Tol* может включать пару *(интел. система, динамич. физич .объект)* и, в силу рефлексивности отношения, пару *(динамич. физич. объект, интел. система)*.

Пусть *S* – сортовая система вида *(St, P, Gen, Tol)*. Тогда произвольная упорядоченная четверка *Ct* вида *(X, V, tp, F)* в [1, 8] называется *концептуально-объектной системой*, согласованной с сортовой системой *S* ⇔ когда выполняются следующие условия:

(1) *X, V* – счетные непересекающиеся множества символов; *tp* – отображение $X \cup V \to Tp(S)$;

(2) *F* – непустое подмножество множества *X*, для каждого $r \in F$ цепочка *tp(r)* начинается с подцепочки «{(« и заканчивается подцепочкой «)}»;

(3) *St* – непустое конечное подмножество множества *X*, и для любого $s \in St$ выполняется соотношение $tp(s) = \uparrow s$;

(4) $\{v \in V \mid tp(v) = [сущ]\}$ - счетное множество.

Множество *X* называется *первичным информационным универсумом*, элементы множеств *V* и *F* называются соответственно *переменными и функциональными символами*. Если элемент $d \in X \cup V$, $tp(d) = t$, то будем говорить, что *t* – *тип* элемента *d*.

Элементы множеств *X* и *V* интерпретируются как элементарные блоки, из которых (и нескольких

служебных символов) будут строиться семантические представления предложений и дискурсов. Например, *X* может включать элементы *город, отгрузка1, 125, 3/тонна, зелен, контейнер1, Столица, Вес, Цена, Часть, Элемент-множества, V* может содержать символы *x1, y3, z12, F* может включать элементы *Столица, Вес, Цена*.

Определение 1. Пусть *S* – произвольная сортовая система вида *(St, P, Gen, Tol)*, где *St* – множество сортов, *P* – выделенный сорт «смысл сообщения», *Gen* – отношение общности на *St, Tol* – отношение совместимости на *St* (см. [1, 8]). Тогда сортовую систему *S* будем называть **аспектно-ориентированной** ⇔ когда

(1) *St* включает выделенные, попарно различные сорта *техн.устр, физ.об;*

(2) (*физ.об, техн.устр*) ∈ *Gen*;

(3) $\{u \in St \mid (P, u) \in Gen\} \cap \{физ.об, техн.устр\} = \varnothing$.

Сорта *техн.устр* и *физ.об* интерпретируются как обозначения понятий «техническое устройство» и «физический объект».

Определение 2. Пусть *S* – произвольная аспектно-ориентированная сортовая система, *Ct* – концептуально-объектная система вида *(X, V, tp, F)*, согласованная с сортовой системой *S*, где *X* – множество символов (первичный информационный универсум), *V* и *F* – множества *переменных* и *функциональных символов* соответственно. Если элемент $d \in X \cup V$, $tp(d) = t$, то будем говорить, что *t* — тип элемента *d*. Тогда упорядоченная пятерка вида *Ctmw = (X, V, tp, F, Qf)* называется **слабо размеченной концептуально-объектной системой**, согласованной с сортовой системой *S* ⇔ когда выполняются следующие условия:

(1) X\F включает подмножество $Qf = \{r_1, ..., r_{11}\}$, $n = 11$, где

r_1 = описание_структуры,
r_2 =описание_характеристик,
r_3 =описание_работы,
r_4 =описание_функций,
r_5 =описание_назначения,
r_6=описание_применения,
r_7 =описание_принадлежности,
r_8 =описание_различий,
r_9 =описание_общих_характеристик,
r_{10} =описание_особенностей,
r_{11} =описание_возможностей;

(2) $tp(r_1)=tp(r_2)=...tp(r_{11}) = \{(физ.об, P)\}$.

Определение 3. Пусть *S* – произвольная аспектно-ориентированная сортовая система, *Ctmw* – слабо размеченная концептуально-объектная система вида *(X, V, tp, F, Qf)*, согласованная с *S*. Тогда упорядоченный набор *Cobs* вида *(X, V, tp, F, Qf, Chr, Qnf, Fgn)* будем называть **размеченной концептуально-объектной системой**, согласованной с сортовой системой *S* ⇔ когда выполняются следующие условия:

1. Набор *(X, V, tp, F, Qf)* является слабо размеченной концептуально-объектной системой, согласованной с сортовой системой *S*;

2. *Chr* – выделенное конечное подмножество множества унарных функциональных символов *F[1]* (интерпретируется как множество характеристик объектов заданной предметной области);

3. *Qnf* – конечное подмножество множества *F[1] \ Chr* (смысл элементов этого множества заключается в представлении характеристик, не принадлежащих объекту интереса поискового запроса);

4. Пусть *Concepts* – множество всех таких *d* из *X,* что тип *tp(d)* начинается с символа ↑(т.е. *d* – обозначение понятия).Тогда *Fgn* – это функция, ставящая в соответствие произвольному *s* из *Concepts* упорядоченную четверку *Gn = (Ge, Concr, Syn, An),* где *Ge* – множество обобщающих понятий объекта интереса поискового запроса, *Concr* – множество конкретизирующих понятий объекта интереса поискового запроса, *Syn* – множество синонимов (на концептуальном уровне) объекта интереса поискового запроса, *An* – множество антонимов объекта интереса поискового запроса, причем *Ge, Concr, Syn, An* – это подмножества множества *Concepts*. Функцию *Fgn* будем называть *детерминантом концептуального окружения*; если *s* – элемент множества *Concepts,* то упорядоченная четверка *Gn(s) = (Ge, Concr, Syn, An)* будет называться *концептуальным окружением термина s*.

Метод построения семантического расширения поискового запроса

Рассмотрим шаги, необходимые для построения семантического расширения поступившего поискового запроса на основе предлагаемой математической модели.

На первом шаге необходимо проанализировать поступивший на вход поисковый запрос *ω* с целью определения его типа. Для этого необходимо использовать определение слабо размеченной

концептуально-объектной системы вида *Ctmw=(X, V, tp, F, Qf)*, а именно, установить соответствие поискового запроса ω одному из элементов множества $Qf = \{r_1, ..., r_{11}\}$. Например, поисковому запросу ω = «Каковы характеристики платформы J2EE?» будет соответствовать реляционный символ $r_2 \in Qf \mid tp(r_2) = \{(физ.об, P)\}$, что означает, что данный запрос имеет тип «*Описание_характеристик*». После того, как определен тип вопроса, необходимо выделить первостепенный и второстепенный объекты интереса поискового запроса ω. Первостепенным объектом интереса будет являться ω_1 = «характеристика», а второстепенным ω_2= «платформа J2EE».

После определения основных характеристик запроса можно переходить к созданию множества вторичных поисковых запросов, порождаемых запросом ω, т.е. к построению семантического расширения входного запроса. Построение данного множества происходит на основании размеченной концептуально-объектной системы *Cobs* вида *(X, V, tp, F, Qf, Chr, Qnf, Fgn)*. Расширение поискового запроса происходит при помощи детерминанта концептуального окружения *Fgn*. Таким образом, необходимо построить набор

$Gn_{\omega_1} = (Ge_{\omega_1}, Concr_{\omega_1}, Syn_{\omega_1}, An_{\omega_1})$.

Для ω_1 = «характеристика» Gn_{ω_1} будет включать следующие элементы:

Ge_{ω_1} = {отзыв, рекомендация},

$Concr_{\omega_1}$ = {описание},

Syn_{ω_1} = {портрет, описание}, $An_{\omega_1} = \varnothing$.

Как только расширенное множество запросов сформировано, оно передается в традиционную поисковую систему, возвращающую множество релевантных документов. При анализе документов будут использоваться такие компоненты размеченной концептуально-объектной системы, как множества *Chr* для определения степени соответствия информации в документе характеристикам, присущим заданному объекту интереса, и *Qnf* для фильтрации документов, содержащих нерелевантную информацию. Кроме того, для фильтрации нерелевантных документов используется множество антонимов $An\omega_1$ из концептуального окружения термина ω_1. Предполагается, что данные множества формируются в зависимости от предпочтений пользователя, т.е. в зависимости от его поведения и выбора тех или иных результатов поиска.

Пример построения семантически преобразованного множества поисковых запросов

Проиллюстрируем на примере построение сначала концептуального окружения термина *s* из отдельно взятого поискового запроса на естественном языке и затем – семантического расширения запроса. Пусть задан запрос ω = «Каковы особенности компьютера MacBook Pro?». Данный поисковый запрос относится к типу (10), описанному ранее, и соответствует реляционному символу *Описание_особенностей*. Первостепенным объектом интереса данного запроса являются особенности определенного объекта, в данном случае – компьютера MacBook Pro, а не сам компьютер. Пусть *s* = *«особенности»*, тогда для определения лексикографического окружения нам необходимо привести данное слово к нормальной форме (именительный падеж, единственное число, т.к. это существительное). Информация для преобразования такого типа традиционно содержится в лингвистической базе знаний, описание которой не затрагивается в данной работе, однако наличие таковой необходимо для реализации предлагаемого подхода. Нормализованный термин *s'*= «*особенность*», в таком случае *Gn(s')* = ({*свойство, черта*},{}, {*непохожесть, отличие*}, {*сходство, похожесть*}).

Как видно из построенного концептуального окружения, данный термин не имеет конкретизирующих понятий. Построение семантического расширения поискового запроса осуществляется следующим образом: сначала добавляется первоначальный поисковый запрос, затем поисковый запрос, использующий синонимы объекта интереса, затем поисковый запрос с конкретизирующими понятиями и, наконец, поисковый запрос, построенный с использованием обобщающих понятий термина *s*. Следует отметить, что построение концептуального окружения проводилось для нормализованного термина *s'*. Для построения грамматически равносильных поисковых запросов необходимо выполнить денормализацию лексикографического окружения и привести его в ту форму, в которой находился термин *s* в поисковом запросе, т.е. *Gn(s)* = ({*свойства, черты*},{},{*непохожести, отличия*},{*сходства, похожести*}). Результирующее множество поисковых запросов будет выглядеть так: *Wse* = {«Каковы особенности компьютера MacBook Pro?», «Каковы непохожести компьютера MacBook Pro?», «Каковы отличия компьютера MacBook Pro?», «Каковы свойства компью-

тера MacBook Pro?», «Каковы черты компьютера MacBook Pro?»}.

Однако, несмотря на проведенные преобразования, результат работы поисковой системы не будет удовлетворительным из-за вопросительной структуры поискового запроса, которая редко встречается в документах, содержащих описания подобных характеристик заданного объекта. Поэтому необходимо выполнить стемминг поисковых запросов.

Следует пояснить, что в традиционном поиске понятие стемминга термина поискового запроса означает сохранение лишь основы слова, чтобы избежать зависимости от различных словоформ, встречающихся в разных документах. Для поискового запроса в целом стеммингом будет являться сокращение данного запроса до семантически значимых составляющих, т.е. до первостепенных и второстепенных объектов интереса. В представленном примере множество *Wse* примет следующий вид: *Wse'* = {«MacBook Pro особенности», «MacBook Pro непохожести», «MacBook Pro отличия», «MacBook Pro свойства», «MacBook Pro черты»}. Изменение порядка и количества слов позволяет с наибольшей точностью найти документы, содержащие необходимую информацию. Элементы представленного множества поисковых запросов с большей вероятностью встречаются в различных документах, что позволяет избежать ограничения лишь по документам, содержащим описание непосредственно «особенностей» данного объекта.

Приведем пример результатов работы поисковой системы (в данном случае — Яndex) для первоначального поискового запроса и для множества сгенерированных запросов:

Таблица 1.

Сравнение результатов выдачи поисковой системы

w = «Каковы особенности компьютера MacBook Pro?»	Wse' = {«MacBook Pro особенности», «MacBook Pro непохожести», «MacBook Pro отличия», «MacBook Pro свойства», «MacBook Pro черты»}
re: Store — Apple MacBook Pro 13″ Ноутбук **MacBook Pro 13″**, блок питания MagSafe, шнур питания, адаптер, диски с ПО, салфетка для очистки экрана, документация. Отличительные **особенности**…	**Apple - MacBook Pro - Спецификации 15-дюймовой модели** **MacBook Pro** имеет следующие **особенности** для снижения воздействия на окружающую среду: корпус из алюминия и стекла, пригодный для вторичной переработки; экран со светодиодной подсветкой, не содержащий ртути
Ремонт ноутбуков MacBook от Apple, срочный ремонт MacBook Pro… Здравствуйте, у меня на **macbook pro** сначала перестал записивать дисковод, а потом там остался диск и он не выдает его обратно, какова ориентировочная стоимость ремонта? Ответ ...	**Apple - MacBook Pro - Производительность - Скоростные процессоры…** В **отличие** от систем, где модули памяти подключены к процессору через отдельный контроллер, в новом **MacBook Pro** используется интегрированный контроллер памяти, с которым можно подключать память напрямую к процессору.
	Apple MacBook Pro -- Обзоры -- mobi.ru Главная **черта** внешности **MacBook Pro** — минимализм. Ничего лишнего, ни единого декоративного элемента.
Мидис Запорожье:: Ноутбук Apple MacBook Pro Компьютеры и ноутбук… Всю необходимую информацию о **MacBook Pro** вы найдёте на сайте Apple.com: видеоруководства по iLife, **Mac** OS X, Aperture и другие материалы. Узнайте об **особенностях** нового **MacBook Pro**. Прочтите советы по работе с ним.	**Top List: Ноутбуки - обзор Apple MacBook Pro. Характеристика. Описание** Дорожные **свойства**. Первое, что ощущается, когда берёшь **MacBook Pro** в руки – это его совсем небольшая толщина.
	Замена жёсткого диска в MacBook Pro - Все о продукции Apple Замена жёсткого диска в моём **MacBook Pro** прошла просто замечательно… Каждая из моделей обладает своими особенностями, придающими ноутбукам определенные **свойства**, полезные в тех или иных...

Из результатов, представленных в *таблице 1*, видно, что пользователь получает более широкий набор результатов, удовлетворяющих его запросу. Более того, в результатах не будут присутствовать вопросительные слова. В дальнейшем планируется использовать специализированную базу знаний, содержащую определения объектов интереса поискового запроса и некоторую дополнительную информацию, которая может быть использована при анализе результатов работы поисковой системы.

Построение лингвистической базы знаний

Построение семантических представлений аспектно-ориентированных поисковых запросов должно поддерживаться лингвистической базой знаний. Лингвистическая база знаний (ЛБЗ) необходима для хранения информации о грамматических свойствах слов, для определения семантического контекста того или иного слова, для построения взаимосвязей между словами, а также для добавления определений значений слов и построения иерархии понятий. Для построения такой базы требуется адекватная мате-

матическая модель. В теории К-представлений [1, 8] предлагается математическая модель лингвистической базы данных, которая частично удовлетворяет поставленным требованиям. Тем не менее, предложенная модель не учитывает взаимосвязи между словами, которые являются синонимами, антонимами, меронимами, холонимами, гиперонимами и гипонимами. Для определения данных связей между словами предложенную математическую модель ЛБД необходимо дополнить функцией, ставящей заданному слову в соответствие его расширенное лексикографическое окружение в зависимости от значения и предметной области, которой принадлежит данное слово. Помимо этого, требуется найти такой источник данных, который позволил бы наполнить построенную модель фактическими данными. Существующие в настоящее время коллективно разрабатываемые базы знаний Wiktionary и Wikipedia [9] имеют ряд преимуществ по сравнению с такими традиционными решениями, как WordNet [10] и некоторые его разновидности [9]. Тем не менее, данные источники данных не являются полностью приемлемыми в силу различных структурных и технологических ограничений. В настоящее время ведется работа над оптимизацией их использования в качестве источника данных, а также преобразования их структуры в наиболее соответствующую разрабатываемой математической модели.

Заключение

Предлагаемый подход не только решает проблемы традиционных поисковых систем, но и позволяет расширить возможности семантического поиска за счет реализации естественно-языкового интерфейса, построения семантически близких, но грамматически отличающихся запросов, а также возможности использовать базу знаний, содержащую информацию о семантических единицах.

Введенное формальное определение размеченной концептуально-объектной системы, по сравнению с введенным В.А. Фомичевым в теории К-представлений понятием концептуально-объектной системы (см. [1, 8]), позволяет:

✦ конструировать иерархию понятий (а не только сортов) по степени их общности;

✦ ставить в соответствие понятию его лексикографическое окружение;

✦ определять обобщающие, уточняющие и аналогичные семантически, но отличающиеся синтаксически понятия (с помощью функции «детерминант концептуального окружения понятия»);

✦ выделить подкласс информационных единиц с отрицательным значением.

Предложенная формальная модель полностью поддерживает процесс преобразования поискового запроса к расширенному виду, связанный с построением множества семантически близких поисковых запросов. Представляется, что продолжение работы в данном направлении позволит расширить множество анализируемых вопросов, повысить качество анализа при помощи дополнительных критериев, а также построить систему семантически-ориентированного анализа естественно-языковых вопросов, позволяющую значительно повысить семантическую релевантность результатов работы традиционных систем поиска по ключевым словам. ■

Литература

1. Фомичев, В. 2005. Формализация проектирования лингвистических процессоров. МАКС Пресс. Москва.
2. Кириллов, А. 2009. Поисковые системы: компоненты, логика и методы ранжирования. Бизнес-информатика № 4(10). С. 51—59. Москва.
3. Halpin, H and Lavrenko, V. 2009. Relevance Feedback Between Hypertext and Semantic Search. Proc. Conference WWW2009 (April 20-24, 2009, Madrid, Spain).
4. Meij, E, Mika, P and Zaragoza, H. 2009. Investigating the Demand Side of Semantic Search through Query Log Analysis. Proc. Conference WWW2009 (April 20-24, 2009, Madrid, Spain).
5. Fernandez, M and Lopez, V. 2009. Using TREC for cross-comparison between classic IR and ontology-based search models at a Web scale. Proc. Conference WWW2009 (April 20-24, 2009, Madrid, Spain).
6. Dali, L, Rusu, D and Fortuna, B. 2009. Question Answering Based on Semantic Graphs. Proc. Conference WWW2009 (April 20-24, 2009, Madrid, Spain).
7. Akbik, A and Broth, J. 2009. Wanderlust: Extracting Semantic relations from Natural Language Text Using Dependency Grammar Patterns. Proc. Conference WWW2009 (April 20-24, 2009, Madrid, Spain).
8. Fomichov, V. 2010. Semantics-Oriented Natural Language Processing: Mathematical Models and Algorithms. Springer: New York, Dordrecht, Heidelberg, London.
9. Zesch, T, Müller, C and Gurevych, I. 2008. Extracting Lexical Semantic Knowledge from Wikipedia and Wiktionary. Proc. Conference LREC 2008 (Marrakech, Morocco).
10. Fellbaum, C and Miller, G. 1998. WordNet. An electronic lexical database. Cambridge, MA: MIT Press; 1998. 422 p.

✦

DEVELOPMENT OF A MODEL AND AN ALGORITHM FOR INFORMATION FLOWS MANAGEMENT IN SITUATION CENTERS

I. Vakhmyanin, N. Ilyin, E. Novikova

Annotation

The article contains description of development of mathematical model for data flows in situational control centers. Optimal algorithm of data flow control is described also. New integrated software and hardware solution VIRD (an acronym for «Visualization of Information using Distributed Displays») was developed using this optimal algorithm. This solution was installed in the Situational Control Center of Administration of Penza Region.

Key words: visualization, situational center, situational room, control center, control room, data flow model, BI, business analysis, data flow model, control software, VIRD.

✦

METHOD OF THE DECISION OF DISTINCTION OF DIGRAPHS PROBLEM ON THE BASE OF COMPLEXITY

V.A. Kokhov, V.V. Kokhov

Annotation

The method of the decision of distinction of digraphs problem is offered. The basis of this method is defined on matrix model of complexity, which takes into account the quantitative and qualitative characteristics of digraph fragments. The model for the first time allows to calculate the importance of each digraph fragment of digraph in its total complexity. The results of the decision of problems of distinction and definition of similarity for digraphs are given.

Key words: Digraph; complexity of digraphs; distinction of digraphs problem; model of complexity; similarity of digraphs; arrangement of paths.

✦

COMPONENT ARCHITECTURE WITH RUNTIME TYPE DEFINITION

E. Gringkrug, A. Shakurov

Annotation

The article focuses on the component-based approach to software design and development. By analyzing the main ideas of this approach, their implementation in the major modern component technologies, their limitations and promising lines of development we suggest a new component architecture which extends capabilities of the existing component technologies. The main principles for building such and architecture are described.

Key words: component, component-based development, interface, implementation.

✦

ONTOLOGICAL ENGINEERING IN SYSTEM DESIGN OF CORPORATE INFORMATION SYSTEMS

N. Lychkina, A. Idiatullin

Annotation

The article deals with a through service-oriented design of integrated information-analytical systems based on architectural approach using ontological modeling framework of the Enterprise architecture. It describes the complex ontological models of the enterprise: a model of meta-ontology, model of the business environment and the ontology of strategic information systems - in accordance with the scheme of an enterprise architecture framework, the ontology of strategic decision making.

Key words: enterprise information systems, decision support system, ontology, semantic network of models.

✦

INFORMATION TECHNOLOGIES FOR FINANCIAL RISK MANAGEMENT

S. Avdoshin, E. Pesotskaya

Annotation

Most of financial operations are used in uncertainty and require risk management. Risk management system should consist of the proved technique and methodology risks management and technology base supported by IT system. The choice of program software for risk management seems to be a hard task for the company staff and requires involvement of professional specialists.

Key words: Information Technologies, financial risks, risk-management, software.

✦

PREDICTION OF BANKRUPTCY PROBABILITY DEPENDING ON VARIATION OF FINANCIAL COEFFICIENTS IN DYNAMICS

T. Bogdanova, J. Alekseeva

Annotation

System of logistic regression models is suggested which takes into account variation of financial coefficients in dynamics. Besides it comparative analysis of predictive accuracy is performed for the system of models on the basis of learning and control samples of manufacturing companies.

Key words: logistic regression model, financial performance of the company, prediction of bankruptcy probability, learning sample, control sample, variation of financial coefficients in dynamics.

✦

ABOUT A NEW APPROACH TO SEMANTIC TRANSFORMATION OF NATURAL LANGUAGE REQUESTS OF SEARCH SYSTEMS

A. Kirillov, V. Fomichov

Annotation

The article considers the existing approaches to information search and their main disadvantages. A new approach to information retrieval based on semantic transformation of search queries is elaborated. According to the proposed approach, the classes of natural language queries are selected, a formal model of a problem-oriented system of primary items of conceptual level and a method of constructing a semantic expansion of the search query are proposed, an application of this method is illustrated.

Key words: semantic search, semantic analysis, theory of K-representations, natural language requests, search systems.

ЖУРНАЛ «БИЗНЕС-ИНФОРМАТИКА» ОСУЩЕСТВЛЯЕТ РАЗМЕЩЕНИЕ РЕКЛАМНЫХ И РЕКЛАМНО-ИНФОРМАЦИОННЫХ МАТЕРИАЛОВ

Расценки:

Обложка: 2, 3, 4 страница обложки, полноцветная печать, полоса 210×290 мм (А4) – 40 тыс. руб.

Текстовый блок, чёрно-белая печать:
- ✦ полоса – 20 тыс. руб.;
- ✦ 1/2 полосы – 15 тыс. руб.;
- ✦ 1/4 полосы – 10 тыс. руб.;
- ✦ меньший объём – 7 тыс. руб.

Вставка (4 полосы, полноцветная печать – 60 тыс. руб.).

Рекламно-информационный блок (8 полос, полноцветная печать) – 80 тыс. руб.
Рекламно-информационный блок (16 полос, полноцветная печать) – 90 тыс. руб.

Корпоративный специальный выпуск – по договоренности.

Материалы принимаются с учётом следующих параметров:
- ✦ дообрезной формат – 215×300 мм;
- ✦ обрезной формат – 210×290 мм;
- ✦ поле набора полосной рекламы – 190×270 мм – с отступом от границ обрезного формата по 10 мм с каждой стороны;
- ✦ файл TIF, EPS, PDF – разрешение не менее 300 dpi.

Тематичекие рубрики журнала «БИЗНЕС-ИНФОРМАТИКА»

№	Рубрика
1	Математические модели социальных и экономических систем
2	Програмная инженерия
3	Анализ данных и интеллектуальные системы
4	Математические методы и алгоритмы решения задач бизнес-информатики
5	Моделирование и анализ бизнес-процессов
6	Информационные системы и технологии в бизнесе
7	Электронный бизнес
8	Интернет-технологии
	Дополнительные рубрики вне номенклатуры
9	Тематические обзоры
10	Правовые вопросы бизнес-информатики
11	Стандартизация, сертификация, качество, инновации
12	Дискуссионный клуб / Опыт бизнеса

К СВЕДЕНИЮ АВТОРОВ

Редакция просит авторов при оформлении статей и тематических обзоров придерживаться следующих правил и рекомендаций:

1. Предоставляемый авторами материал должен соответствовать рекомендуемой структуре статей журнала.

2. Статья направляется в редакцию в электронном виде (в формате MS WORD версия 2003) и в виде бумажной копии, распечатанной на одной стороне листов А4. Первая страница оригинала подписывается всеми авторами статьи.

3. Ориентировочный объём статьи, предлагаемой к публикации, – 20–25 тыс. знаков (с пробелами) или 30–35 тыс. знаков – для обзорных статей по направлениям.

4. Кегль набора – 12 пунктов с полуторным интервалом. Нумерация страниц – вверху по центру. Поля: левое – 2,5 см, верхнее, нижнее и правое – по 1,5 см.

5. При наборе выключных и строчных формул должен быть использован редактор формул MS Equation. В формульных и символических записях греческие (русские) символы, а также математические функции записываются прямыми шрифтами, переменные аргументы функций в виде английских (латинских) букв записываются наклонным начертанием (курсивом), например, «cos *a*», «sin *b*», «min», «max».

6. Формулы, таблицы и сноски (не концевые) оформляются стандартными средствами редактора MS WORD. Нумерация формул, рисунков и таблиц – сквозная, по желанию авторов допускается двойная нумерация формул с указанием структурного номера раздела статьи и – через точку – номера формулы в разделе.

7. Рисунки (графики, диаграммы и т.п.) оформляются средствами Word, Excel, Illustrator. Ссылки на рисунки в тексте обязательны и должны предшествовать позиции размещения рисунка. Допускается использование графического векторного файла в формате wmf/emf или cdr v.10. Фотографические материалы предоставляются в формате TIF или JPEG с разрешением не менее 300 dpi.

8. Библиографический список составляется в соответствии с требованиями ГОСТ. Нумерация библиографических источников – в порядке цитирования. Ссылки на иностранную литературу – на языке оригинала без сокращений.

Структура статей строится по правилам, рекомендованным журналом «Бизнес-информатика».

Плата с аспирантов за публикацию рукописей не взимается.

Журнал «БИЗНЕС-ИНФОРМАТИКА» Рекомендуемая структура статей

Журнал публикует исследовательские научные статьи, размещаемые в рубриках журнала, тематические обзоры, отражающие современное состояние проблем в области бизнес-информатики и сообщения, размещаемые в рубриках «Дискуссионный клуб» и «Опыт бизнеса».

Титульный лист рукописи начинается с указания Ф.И.О. авторов публикации с обязательным указанием учёной степени, учёного звания, должности, основного места работы и e-mail. Титульный лист должен быть подписан всеми авторами статьи.

I. Исследовательские научные статьи (для размещения в тематических рубриках)

Редколлегия рекомендует авторам после названия статьи приводить ***аннотацию***, в которой излагается краткое содержание статьи, её основные результаты и область применения. Авторам рекомендуется структурировать статью, выделяя***введение***, содержащее описание проблемы или задачи, обзор существующих подходов или методов решения, их недостатки, и основную цель статьи; ***постановку задачи***, включающую допущения и ограничения; ***содержательную часть*** статьи, в которой предлагаемые решения должны быть аргументированы и сравниваться с существующими подходами или решениями; ***заключение***, содержащее краткое изложение новых результатов, полученныхв статье и область их применения; ***библиографический список***, оформленный в соответствии с ГОСТ. Текст статьи должен содержать нумерованные ссылки на все указанные библиографические источники. Структурирование статьи и нумерация её разделов проводится по усмотрению авторов.

Возможный вариант структуры статьи:

- Ф.И.О;
- учёная степень, учёное звание, должность, основное место работы, e-mail;
- название статьи.
- аннотация;
- 1. Введение.
- 2. Постановка задачи.
- 3. Основная содержательная часть статьи.
- 4. Экспериментальные результаты (опционально).
- 5. Заключение.
- 6. Библиографический список.

II. Тематические обзоры по направлениям

Редколлегия рекомендует авторам структурировать обзор, выделяя аннотацию, содержащую тематику, краткое содержание обзора и область применения;***введение***, в котором даётся краткий исторический обзор тематики;***содержательную часть*** обзора с критическим анализом существующих направлений;***заключение***, в котором отражаются перспективы развития в рамках обозреваемой тематики и наиболее интересные направления с точки зрения научных и практических разработок и методов; ***библиографический список***, оформленный в соответствии с ГОСТ.

Текст обзора должен содержать нумерованные ссылки на все указанные библиографические источники. Структурирование обзора и нумерация его разделов проводится по усмотрению авторов.

Возможный вариант структуры обзора:

- Ф.И.О;
- учёная степень, учёное звание, должность, основное место работы, e-mail;
- название обзора;
- аннотация;
- 1. Введение.
- 2. Основная содержательная часть обзора.
- 3. Заключение.
- 4. Библиографический список.

Редколлегия журнала проводит обязательное рецензирование рукописей. Статья принимается к публикации только после получения положительного заключения рецензента и одобрения на заседании редакционной коллегии журнала.