

АВТОМАТИЗИРОВАННАЯ ИНФОРМАЦИОННАЯ УПРАВЛЯЮЩАЯ СИСТЕМА УНИВЕРСИТЕТА

С.А. Яблонский, к.т.н.; А.М. Сухоногов; А.А. Темплинг; О.А. Самонина
(Петербургский государственный университет путей сообщения, serge_yablonsky@hotmail.com)

В работе приводится описание автоматизированной информационной системы управления Петербургского государственного университета путей сообщения. Рассматривается программная платформа системы, состоящая из СУБД Oracle 10g, Oracle SSO, Oracle Internet Directory, сервера приложений Oracle AS 10g и порталного решения Oracle Portal. Кратко описываются основные компоненты системы и задачи, решаемые ими. Автоматизированная система обеспечила единое информационное пространство университета, доступное как сотрудникам, так и студентам для совместного решения широкого круга задач – от контроля успеваемости студентов до расчета численности штатов.

Ключевые слова: автоматизированная информационная система управления университетом, СУБД Oracle 10g, Oracle SSO, Oracle OID, Oracle AS 10g, Oracle Portal.

Модернизация современной российской системы образования обусловлена, с одной стороны, присоединением России к Болонскому процессу, где преобладают европейские стандарты, с другой стороны, необходимостью повышения качества естественно-научного и инженерного образования в России для обеспечения его соответствия современным запросам рынка труда и технологий.

Министерство образования и науки РФ сформировало требования, которым должны отвечать университетские системы управления учебным процессом (<http://www.informika.ru/text/inftech/iais/>). Отмечается, что разработка этих систем должна проводиться на единой технологической базе на основе единого ядра в рамках стандартов единого информационного пространства вуза.

Анализ существующих университетских АСУ [1] позволяет выделить:

- системы административного управления (*Education Administrative Systems*);
- системы управления учебными ресурсами (*Higher Education Content Management*);
- системы дистанционного обучения (*Higher Education E-Learning Systems*);
- библиотечные системы (*Higher Education Library Management*);
- системы поддержки обучения и исследований (*Higher Education Learning, Research and Intellectual Assets*);
- системы управления высшим образованием (*Higher Education Management*).

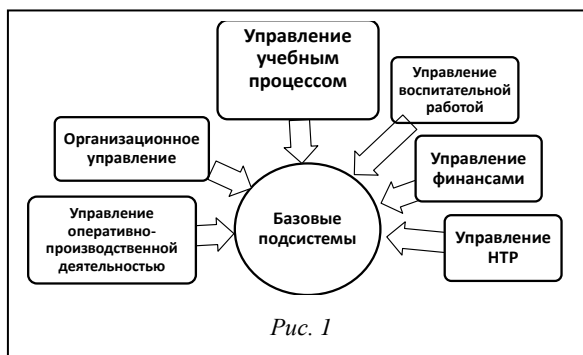
Корпорация Oracle известна как один из ведущих разработчиков подобных систем [1]. Однако часть готовых решений Oracle, к сожалению, не учитывает особенности принятых в российской высшей школе бизнес-процессов обучения и контроля обучения, что определяет необходимость разработки отдельных подсистем специально для российских университетов.

Для автоматизированной информационной системы управления (АИСУ) Петербургского государственного университета путей сообщения (ПГУПС) [2–5] были выбраны СУБД Oracle 10g, инфраструктурные решения Oracle (SSO – Single-

Sign On, OID – Oracle Internet Directory), сервер приложений Oracle AS 10g и порталное решение Oracle Portal.

До реализации проекта АИСУ ПГУПС в университете в большом количестве были распространены локальные решения на базе офисных продуктов Microsoft. Причем каждое подразделение самостоятельно собирало, актуализировало и обрабатывало необходимые ему данные. АИСУ позволила создать единое информационное пространство университета, доступное как сотрудникам, так и студентам для совместного решения широкого круга задач.

Основные процессы управления и соответствующие подсистемы, реализованные в АИСУ ПГУПС, показаны на рисунке 1.



АИСУ ПГУПС решает следующие задачи:

- автоматизация сбора и передачи информации об успеваемости студентов;
- автоматизация процессов контроля успеваемости;
- оперативное оповещение руководства всех уровней об успеваемости студентов и о других показателях работы университета;
- оперативный контроль процесса обучения и повышения квалификации студентов и сотрудников университета;
- создание единой защищенной централизованной корпоративной базы данных ПГУПС;
- улучшение качества информационного обслуживания студентов и преподавателей ПГУПС;

- накопление данных по информационным картам студентов и преподавателей;
- накопление данных по учебным планам и другим руководящим документам, определяющим учебный процесс в университете;
- уменьшение доли ручного труда;
- устранение повторного ввода одних и тех же данных об абитуриентах, студентах, аспирантах, докторантах и преподавателях университета;
- унификация данных, необходимых для управления университетом;
- автоматизация формирования отчетности;
- автоматизация информационного сопровождения студента от момента подачи заявления в приемную комиссию до выпуска из вуза.

Согласно требованиям к АИСУ со стороны руководства не подлежат автоматизации процессы управления финансово-хозяйственной деятельностью, запрещается непосредственная работа АИСУ с кадровой системой сотрудников университета (автоматизирован только обмен данными в ограниченном объеме).

Основными пользователями АИСУ являются:

- на уровне ректората – ректор и проректоры;
- на уровне управлений – руководители и заместители руководителей, ключевые сотрудники отделов;
- на уровне факультетов – деканы, заместители деканов и секретари;
- на уровне кафедр – секретари (ответственные), заведующие кафедрами, заместители заведующих.

В настоящее время разработаны следующие основные подсистемы АИСУ ПГУПС (рис. 2).

- АРМ приемной комиссии. Позволяет организовать совместную работу регистраторов приемной комиссии, сотрудников отдела договорных форм обучения, сотрудников управления международных связей, сотрудников отборочных комис-

сий деканатов, ответственных работников приемной комиссии, удаленную работу сотрудников филиалов (Мурманского, Великолукского, Петрозаводского) с использованием Web-браузера по протоколам *HTTPS/SSL*.

- АРМ учебного управления. Решает задачи ведения рабочих учебных планов специальностей, графика контроля успеваемости, работы с ведомостями успеваемости, планирования численности штата ППС.

- АРМ кафедры. Обеспечивает первичный ввод данных об успеваемости, ввод и ведение индивидуальных планов преподавателей и позволяет оперативно формировать сводную отчетность.

- АРМ факультета. Осуществляет мониторинг процесса обучения, обеспечивает работу с ведомостями дополнительной сдачи, ведение информационных карт студента и экрана успеваемости. Строится сводная отчетность по успеваемости, в том числе и в сравнении с другими факультетами (формируются рейтинги).

- АРМ отдела кадров студентов. Реализует ограниченный документооборот приказов по контингенту студентов университета с автоматическим изменением корпоративной базы данных университета. Обеспечивает формирование и регистрацию различных справок по контингенту студентов.

- АРМ договорных форм обучения. Решает задачи учета данных договоров студентов с возмещением затрат на обучение, учета оплат и задолженностей студента и других контрагентов.

- АРМ целевой и контрактной подготовки. Решает задачи учета договоров об обучении с третьими организациями и договоров целевых студентов, оплаты по этим договорам. Одной из функций АРМ является ведение журнала трудоустройства выпускников университета.

- АРМ управления международных связей. Решает задачи учета данных об иностранных студентах университета.

Например, для автоматизации контроля успеваемости студентов очной формы обучения создано более 100 АРМ, из них более 60 – на кафедрах и факультетах. Для решения задачи задействованы АРМ учебного управления, факультетов и кафедр. Мониторинг осуществляется с помощью информационных панелей руководителей соответствующих подразделений.

Реализована многозвенная клиент-серверная архитектура АИСУ ПГУПС (рис. 3), в которой для работы клиента достаточно наличия на рабочем месте только Web-браузера (*Internet Explorer* версии 6 и выше, *Firefox* и т.п.). Настройка клиентско-

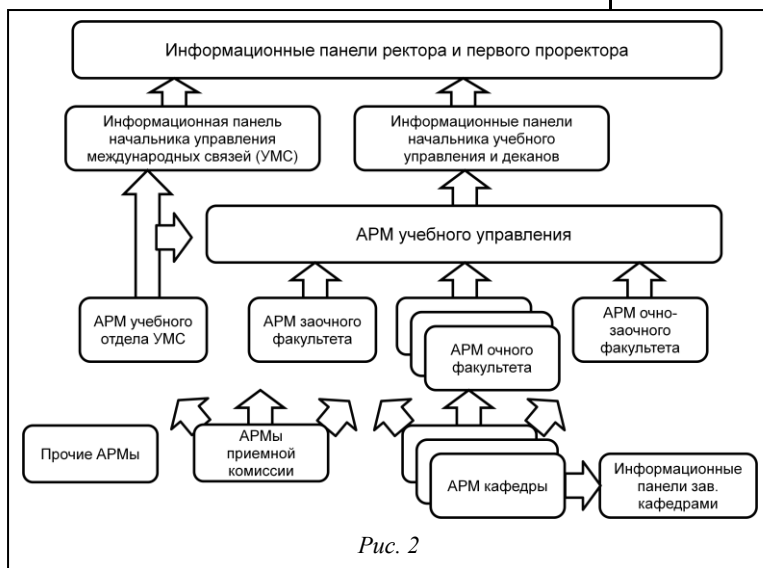
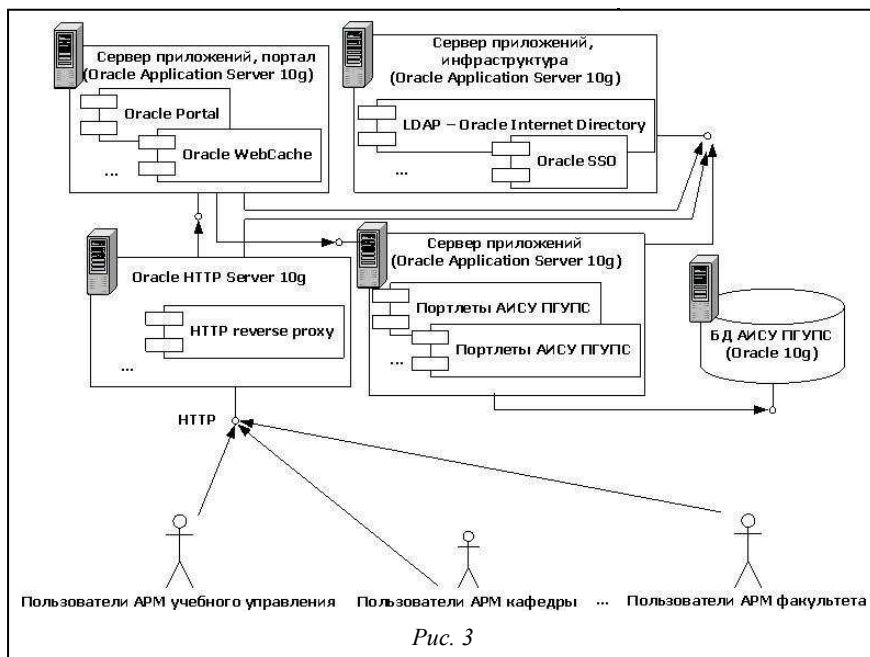


Рис. 2



го рабочего места заключается в определении страницы портала АИСУ ПГУПС как стартовой страницы браузера. Развертывание и обновление программного обеспечения, управление правами доступа осуществляются централизованно на сервере приложений. Таким образом, решается задача минимизации затрат на обслуживание и сопровождение системы.

Для управления доступом пользователей к приложениям используются решения Oracle, поставляемые вместе с сервером приложений:

- LDAP каталог Oracle Internet Directory (OID) для хранения данных о пользователях и назначенных им ролях;
- Oracle SSO для организации единой точки входа во все приложения.

На отдельном сервере приложений Oracle AS 10g развернуты портлеты Oracle Portal 10g. Портлет отвечает за представление и управляет отдельной областью страницы портала. Портлеты взаимодействуют:

- с базой данных АИСУ через пул соединений, позволяющий значительно снизить нагрузку на СУБД при одновременной работе большого числа пользователей;
- с сервером Oracle Internet Directory для получения данных о пользователе и его ролях;
- с сервером приложений портала, который запрашивает контекстно-зависимое содержание области HTML-страницы при ее генерации.

Каждое рабочее место АИСУ представляет набор страниц, защищенных средствами Oracle Portal, на которых развернуты портлеты. Решение позволяет повторно использовать один и тот же портлет в различных АРМ. Все АРМ АИСУ и корпоративный портал доступны через один URI, такую возможность обеспечивает использование

HTTP-сервера в режиме reverse-прокси.

Портлеты разрабатываются на языке Java с использованием предлагаемых Oracle расширений в среде Oracle JDeveloper 10g. До 2008 г. использовались только возможности Oracle JPKD, а с 2008 г. все разработки ведутся на основе Oracle ADF/JSF с моделью, соответствующей спецификации Sun EJB3. Портлеты, как правило, используются для организации взаимодействия с пользователем и визуализации данных из базы данных. Бизнес-логика приложений реализуется в пакетах PL/SQL непосредственно в СУБД.

Информационной основой АИСУ ПГУПС является корпоративная база данных, используемая всеми разработанными и разрабатываемыми приложениями. В настоящий момент единая модель базы данных описывается на 26 ER-диаграммах и состоит из 222 таблиц. Проектирование осуществляется на концептуальном уровне (без привязки к какой-либо СУБД) и на физическом (объекты модели привязываются к реализации на СУБД Oracle 10g). На каждом этапе поддерживается состояние модели, соответствующее реально работающей базе данных АИСУ. Для хранения исходных текстов программ используется система контроля версий Subversion (SVN).

Для организации взаимодействия АИСУ ПГУПС с другими информационными системами ПГУПС программные интерфейсы (API) АИСУ и всех внешних информационных систем представляются в виде Web-сервисов на основе WSDL (Web Services Description Language). WSDL позволяет сгенерировать API для работы с системой на любом объектно-ориентированном языке программирования (Java, C#, C++, VisualBasic, Object Pascal, PHP и т.д.). За управление доступом к сервису, за аудит, диагностику и безопасность отвечает сервер приложений, в котором сервис разворачивается как одно из приложений. Таким образом, данные АИСУ доступны для автоматизированной обработки внешними приложениями, а данные внешних приложений – в АИСУ. За организацию работы сервисов по обмену данными в настоящее время отвечают специально разработанные портлеты, а обмен данными инициируется оператором.

АИСУ ПГУПС не является коробочным решением и разрабатывается с учетом особенностей и пожеланий подразделений-заказчиков. Для под-

держания постепенно эволюционирующих (в том числе и из-за внедрения АИСУ) бизнес-процессов университета требуется постоянное содержание небольшой команды разработчиков.

При достижении некоторого порогового значения числа пользователей (а в настоящее время их число превышает 150) необходимо создание полноценного *call*-центра для обслуживания и систематизации запросов пользователей.

Внедрение АИСУ позволило избавиться от дублирования данных в различных подразделениях, формализовать и регламентировать их бизнес-процессы, организовать единое информационное пространство для совместной работы подразделений, обеспечить доступ руководства как к первичным, так и к статистическим документам системы. Компания *Oracle* включила АИСУ ПГУПС в список лучших примеров внедрения технологий *Oracle* в вузах России (www.oracle.com/global/ru/academy/spb_gu_transport.doc).

Литература

1. Marti Harris, Michael Zastrocky, Jan-Martin Lowendahl Magic Quadrant for Higher Education Administrative Suites, 2007, Gartner, Inc., September 26, 2007.
2. Яблонский С.А., Сухоногов А.М., Темплинг А.А., Самонина О.А. Автоматизированная информационная система управления Петербургского государственного университета путей сообщения (АИСУ ПГУПС) // Информационная среда вуза XXI века: тр. II Всерос. науч.-практич. конф. Петрозаводск, 2008. С. 177–179.
3. Яблонский С.А., Сухоногов А.М., Темплинг А.А., Самонина О.А. Подсистема приемной комиссии АИСУ ПГУПС // Там же. С. 171–174.
4. Яблонский С.А., Сухоногов А.М., Темплинг А.А., Самонина О.А. Расчет численности штата профессорско-преподавательского состава университета // Там же. С. 175–177.
5. Яблонский С.А., Сухоногов А.М., Темплинг А.А., Самонина О.А. Автоматизация ведения приемной кампании 2009 года на основе АИСУ ПГУПС // Информационная среда вуза XXI века: тр. III Всерос. науч.-практич. конф. Петрозаводск, 2009.

ИНСТРУМЕНТАЛЬНАЯ СИСТЕМА ДЛЯ РЕШЕНИЯ ЗАДАЧ МНОГОКРИТЕРИАЛЬНОГО ВЫБОРА

С.В. Микони, д.т.н.; Д.П. Бураков, к.т.н.; М.И. Гарина, к.т.н.
(*Петербургский государственный университет путей сообщения, svm@sm4265.spb.edu*)

Описывается программная система СВирЬ-Р, предназначенная для решения задач многокритериального выбора на конечном множестве альтернатив. Система реализует основные методы выбора, функционирует автономно, комплектуется с информационными системами общего назначения и может конфигурироваться под заданный класс задач. Она успешно применяется в учебном процессе и научных исследованиях.

Ключевые слова: отбор, упорядочение, классификация, модель выбора, методы выбора, критерий, важность критерия, функция полезности, функция принадлежности, конфигурация системы.

К задачам многокритериального выбора на конечном множестве альтернатив относятся нахождение наилучшего (наихудшего, среднего) объекта, отбор допустимых объектов, упорядочение объектов по предпочтению, а также отнесение объекта к одному из заданных классов [1].

Для решения задач выбора применяются методы многокритериальной оптимизации и классификации. Для учебных и практических приложений разработаны многочисленные программы, решающие частные задачи выбора. Ряд программ этого класса встраиваются в системы хранения и обработки данных с целью поддержки принятия решений. Вместе с тем существует потребность в автономной программной системе, инвариантной относительно частных моделей выбора, в которой задачи выбора могут решаться как по отдельности, так и в любых сочетаниях. Такая система востребована в учебном процессе при освоении всей совокупности основных методов выбора и в научных исследованиях, целью которых является изучение свойств моделей и методов выбора путем

проведения экспериментов. Поскольку универсальная система выбора избыточна по отношению к частным задачам, решаемым на АРМ, полезным свойством системы является возможность конфигурации ее ресурсов под конкретные задачи. Будучи автономной, система многокритериального выбора должна быть способной к обмену данными с информационными системами общего назначения и типовыми СУБД.

В Петербургском государственном университете путей сообщения работа над системой с перечисленными свойствами ведется в течение 10 лет. За этот период было разработано и внедрено в учебном процессе и на практике несколько редакций системы выбора и ранжирования СВирЬ [2]. В настоящей статье описываются свойства последней редакции системы, названной СВирЬ-Р, с учетом возможности ее реконфигурации.

Модель многокритериального выбора

Модель выбора в системе, реализующей различные методы многокритериальной оптимизации

и классификации, должна быть инвариантна по отношению к этим методам. Данному требованию отвечает теоретико-множественная модель [3]:

$$M_b = \langle X, F, Y, H, P, \lambda \rangle. \quad (1)$$

Она содержит:

- множество оцениваемых альтернатив (объектов) X ;
- множество функций F , характеризующих различные свойства объектов;
- области определения и значений Y функций из множества F ;
- множество целей (целевых состояний) H , задаваемых для оцениваемых признаков;
- отношения предпочтения (предикаты) P , определяемые на множествах X, F, Y, H ;
- степень соответствия $\lambda_{i,k}$ альтернативы x_i целевому состоянию h_k .

В отличие от моделей другого назначения в модели выбора основополагающую роль играет отношение предпочтения, представленное в формуле (1) двухместным предикатом P .

При сопоставлении альтернатив x_i и x_k , $x_i, x_k \in X$, истинность двухместного предиката $P_{ord}(x_i, x_k)$ означает, что $x_i \succ x_k$. Если предикат $P_{ord}(x_i, x_k) = \text{false}(0)$, то имеет место либо обратное предпочтение $x_i \prec x_k$, либо равнозначность $x_i \equiv x_k$ сравниваемых альтернатив. Бинарное отношение $R_{ord} \subset X \times X$ является решением задачи выбора на языке бинарных отношений. При критериальном выборе каждой альтернативе x_i ставится в соответствие величина $y_j \in Y$, отражающая j -е свойство альтернативы и вычисляемая с применением функции $f_j: X \rightarrow Y$. Отношение $R_{ord} \subset X \times X$ формируется на основе количественных оценок альтернатив, то есть отношения $R_{cor} \subset Y \times Y$.

Для нахождения наилучшей альтернативы выполняется ранжирование отношения R_{ord} . Оно заключается в присвоении каждой альтернативе $x_i \in X$ целого числа из диапазона $[1, N]$: $R_{num} \subset X \times N$, $R_{num} = \{(x_i, 1), \dots, (x_k, \rho), \dots, (x_s, N)\}$, где $N = |X|$. Число ρ задает рейтинг (ранг) альтернативы – указывает ее место по предпочтению среди остальных альтернатив.

Модель (1) предоставляет всю необходимую информацию для решения задач отбора, упорядочения и классификации объектов в n -мерном пространстве признаков.

Создание и редактирование модели выбора

Отношение $R_{in} \subset X \times F$, которое характеризует исходные данные, представляется таблицей «Объекты/Признаки». Строки таблицы образуют объекты из множества X , а столбцы – признаки из множества F . Клетки таблицы заполняются значениями признаков $y_{ij} = f_j(x_i)$. Таким образом, i -й объект, $i \in \overline{1, N}$, характеризуется вектором значений

$y_i = (y_{i1}, \dots, y_{ij}, \dots, y_{in})$, поэтому второе название таблицы – «Характеристика объектов».

Таблица «Объекты/Признаки», описывающая заданную предметную область (ПрО), либо создается в системе СВИРЬ-Р с клавиатуры, либо вводится из табличного процессора *MS Excel* или из типовой СУБД. При создании с клавиатуры задаются размерность таблицы (число строк и столбцов), имя ПрО, имена объектов и признаков. Данные в таблицу могут заноситься с клавиатуры или из названных систем при условии совпадения имен вводимых объектов и признаков с именами, введенными в таблицу.

При формировании иерархической модели дочерние таблицы создаются на основе признаков таблиц предыдущего уровня иерархии (технология построения сверху вниз). Режим создания таблицы нижнего уровня выбирается правой кнопкой мыши. В появляющемся окне задаются число и имена детализирующих признаков. Структура модели выбора представляется деревом таблиц. Для отображения дерева иерархии в левой части главного окна системы СВИРЬ используется стандартное средство *Tree-View* операционной системы *MS Windows*.

Редактирование модели осуществляется перемещением признаков из одной таблицы в другую с помощью технологии *drag-and-drop*. Она же применяется для подчинения (переподчинения) одной таблицы другой в дереве иерархии.

При необходимости перевода качественных значений признаков в порядковую шкалу выполняется их кодирование числами с возможностью обратного преобразования. Задача сопоставимости объектов решается созданием относительных и удельных признаков. Их значения вычисляются с помощью калькулятора. Помимо его стандартных функций, используются агрегатные функции (минимальное, максимальное, среднее значение, сумма значений, число объектов).

Для формулирования задач выбора в системе СВИРЬ-Р используется контекстно-зависимый интерфейс. В его основу положены два класса задач: многокритериальной оптимизации и классификации. Методы оптимизации, в свою очередь, тоже делятся на два класса: методы векторной и скалярной оптимизации. Методы векторной оптимизации различаются отношением доминирования, реализуемого на векторных оценках объектов: Парето-доминирования, лексиминного и лексикографического (метод приоритета критериев). Первые два метода характеризуются минимальным использованием экспертной информации, но не гарантируют получение линейного порядка объектов при решении задачи упорядочения. Качество решения задачи упорядочения характеризуется показателями доминирования, неразличимости и несравнимости объектов, а также коэффициентом полноты порядка [3].

Методы скалярной оптимизации реализуются с применением обобщающих функций и *матриц парных сравнений* (МПС). Для обобщения однородных признаков, измеряемых в одной шкале, используются функции Байеса и Гурвица, а для обобщения неоднородных признаков применяются аддитивная, две мультипликативные (с прямыми и дополнительными сомножителями) и минимаксная функции.

В рамках многокритериальной классификации решаются задачи отбора (выделения допустимого множества), отбора по образцу (метод мягких притязаний) и отнесения к классам по функциям принадлежности и логическому выводу. При использовании функций принадлежности имеется возможность установления порога принадлежности классу, а также сквозного упорядочения объектов по степени их принадлежности упорядоченным по качеству классам.

Многокритериальная оптимизация на основе МПС реализуется с помощью *системы вычисления приоритетов* (СВП), оформленной в виде динамической библиотеки. Роль системы СВІРЬ-Р в этой задаче – предоставление исходных данных для формирования предпочтений, задание типов матриц и параметров вычисления приоритетов, а также представление результатов.

Требования к признакам

Формирование требований к признакам зависит от решаемой задачи и глубины проникновения в предметную область. При решении задач упорядочения объектов на основе признаков формируются критерии оптимизации. Если ЛПР неизвестно пороговое значение признака, формируется целевой критерий, а если известно – ограничительный [3]. Для целевого критерия задается направление оптимизации (*min* или *max*), а для ограничительного критерия – ограничение значения признака снизу, сверху, в точке или интервале. По умолчанию этим критериям ставятся в соответствие функции полезности – линейные и кусочно-линейные. Если ЛПР известна информация о приращении полезности признака, линейные функции полезности преобразуются в нелинейные. Для выполнения нормализации признака задаются границы его шкалы. Они формируются либо ЛПР, либо автоматически, по выборке объектов.

Критерии могут использоваться одновременно для упорядочения и предварительного отбора объектов. В задачах классификации на шкале признака экспертами задаются функции принадлежности классам. В качестве базовой принята трапециевидальная форма функции принадлежности. Она просто преобразуется в треугольную и прямоугольную формы. Если известна более подробная информация о границах классов, функции принадлежности строятся по точкам.

При использовании обобщающих функций в задачах упорядочения и классификации важность признаков может задаваться напрямую в виде весовых коэффициентов и вычисляться на основе МПС. Формирование МПС и расчет приоритетов выполняются в СВП. Для задания приоритетов в иерархических моделях выбора применяется подсистема управления весами в иерархии. В иерархии с неоднородной структурой используются два принципа выравнивания приоритетов: с обеспечением равноценности критериев в таблицах иерархии и с обеспечением равноценности первичных критериев. Весовые коэффициенты признаков в таблицах иерархии в соответствии с первым принципом рассчитываются сверху вниз, а в соответствии со вторым принципом – снизу вверх. В каждой таблице эксперты могут задавать важность критериев. Для сопоставления экспертных и структурных весов признаков в таблицах верхних уровней иерархии используется специальный показатель. Проблема перераспределения весовых коэффициентов первичных критериев в иерархии решается с применением метода донор–акцептор.

Анализ модели и результатов выбора

В системе СВІРЬ-Р предусмотрены различные табличные и графические средства анализа. Анализ независимости критериев осуществляется с помощью корреляционной матрицы. Результаты многокритериального выбора выводятся в таблицу «Решение». В графической форме представляются: функции полезности и принадлежности классам, графы зависимости признаков и графы доминирования объектов, многокритериальные оценки полезности, вклад в них различных критериев, сравнительные оценки рейтинга объектов, полученных различным способом, деревья иерархии признаков и весовых коэффициентов в иерархии. Для пояснения результатов используются специальные таблицы, а для их различения – цветовая окраска.

Выделение ресурсов под задачи выбора

Под выделяемым ресурсом понимается одна или группа функций системы, использование которых требует специального разрешения. Ресурсы системы делятся на внешние и внутренние. Их выделение осуществляется по-разному. Внешние ресурсы, оформленные в виде динамических библиотек, включаются в комплект поставляемой программной системы. Внутренние ресурсы входят в состав исполняемой программы СВІРЬ-Р. Они делятся на следующие группы: отбор объектов, критерии оценивания, скалярная оптимизация, обобщающие функции, векторная оптимизация, классификация, критерии классификации, иерархия, сервисные функции.

Базовая конфигурация системы СВІРЬ-Р представляет собой совокупность ресурсов, требующихся для проектирования модели выбора, решения контрольной задачи и обмена информацией с процессором *MS Excel*. В качестве контрольной задачи принята задача нахождения рейтинга объектов относительно общей цели. Для выполнения лабораторного практикума по теории принятия решений [4] выделяются ресурсы для применения всех методов выбора с ограничением на размерность модели и на сервисные функции.

Литература

1. Микони С.В. Теория и практика рационального выбора. М.: Маршрут, 2004. 462 с.
2. URL: www.pgups.ru/nauka/mikoni/ (дата обращения: 10.07.2009).
3. Микони С.В. Многокритериальный выбор на конечном множестве альтернатив: учеб. пособие. СПб: Лань, 2009. 273 с.
4. Микони С.В., Гарина М.И. Лабораторный практикум по дисциплине «Теория принятия решений»: учеб. пособие. СПб: ПГУПС, 2009. 97 с.

СИСТЕМА ВЫЧИСЛЕНИЯ ПРИОРИТЕТОВ НА ОСНОВЕ МАТРИЦ ПАРНЫХ СРАВНЕНИЙ

И.С. Киселёв; С.В. Микони, д.т.н.

(Петербургский государственный университет путей сообщения, svm@sm4265.spb.edu)

Описывается система, функционирующая в составе программной системы СВІРЬ-Р в качестве динамической библиотеки. Она предназначена для формирования матриц парных сравнений, анализа их согласованности и вычисления приоритетов сущностей. Алгоритм вычисления приоритетов инвариантен относительно четырех типов матриц, реализуемых в системе. Метод анализа иерархий реализуется совместно с системой СВІРЬ-Р относительно сформированной в ней модели выбора.

Ключевые слова: матрица парных сравнений, экспертное предпочтение, приоритет сущности, согласованность предпочтений, многокритериальная оптимизация.

Усилиями американского математика Томаса Саати метод анализа иерархий (МАИ) получил широкое распространение [1]. Он имеет многочисленные программные реализации. Метод основан на использовании матриц парных сравнений (МПС), содержащих кратные предпочтения. Под кратным предпочтением a_{ij}^{kn} понимается превосходство i -й сущности x_i над j -й сущностью x_j , $x_i, x_j \in X$ в k раз, $k \in [0, 1; 10]$.

Однако, кроме использования в МАИ, МПС имеют самостоятельную ценность. Они могут использоваться в задачах распределения ресурсов, расчета приоритета сущностей, вычисления весовых коэффициентов для обобщающих функций в задачах скалярной оптимизации [2], а также отражать и другие типы предпочтений – ординальные (факты предпочтений), вероятностные предпочтения (в долях от 1), соотношения выигрышей и потерь. Помимо субъективных (экспертных) предпочтений, МПС могут отражать объективные предпочтения, такие как результаты футбольного первенства или товарообмен между хозяйствующими субъектами.

Таким образом, в качестве объекта программной реализации целесообразно использовать не метод анализа иерархий, а его важнейшее звено – МПС как более универсальную модель. Этот принцип реализован в программной системе, названной системой вычисления приоритетов (СВП). В статье излагаются возможности этой системы.

Создание МПС

Программная система СВП реализована в виде динамической библиотеки системы выбора и ранжирования СВІРЬ-Р. Из системы СВІРЬ-Р задаются перечень сопоставляемых сущностей, размерность и тип матрицы. В саму систему возвращается вектор нормированных приоритетов, сформированный на основе МПС. Тип МПС (кратности предпочтений, фактов предпочтений, долей от 1 или отношения «Выигрыши/Потери») может задаваться также и внутри системы СВП.

Содержимое МПС может вводиться напрямую экспертами через клавиатуру либо из табличного процессора *Excel*, содержащего известные предпочтения – экспертные или объективные. Матрица может формироваться автоматически на основе вектора приоритетов, заданного в интервальной шкале. Такая матрица имеет стопроцентную согласованность и в случае количественных предпочтений называется сверхтранзитивной. Пример задания вектора приоритетов (в процентах) для четырех признаков приведен на рисунке 1. В правой части вкладки «Задание» выбираются тип МПС, максимальная кратность предпочтения (по умолчанию 10), а также способ автоматического формирования содержимого матрицы.

Содержимое МПС, полученное вручную или автоматически, фиксируется на вкладке «Предпочтения» (рис. 2).

Ручное задание и редактирование предпочтений осуществляются на правой стороне вкладки.

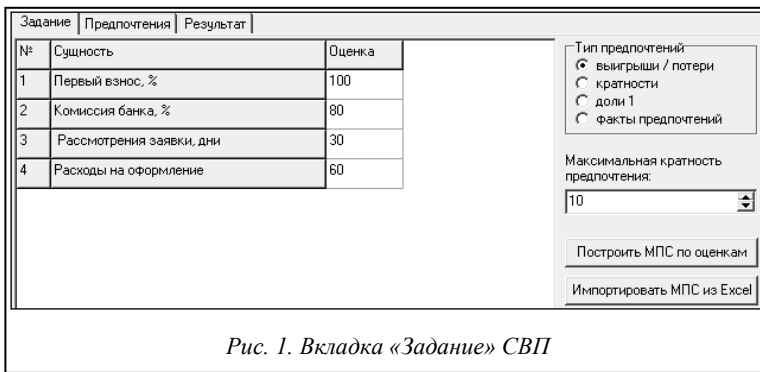


Рис. 1. Вкладка «Задание» СВП

В верхней ее части фиксируются имена сопоставляемых сущностей. Каждому типу матрицы соответствуют три ряда окошек ввода. При вводе предпочтения в выбранное окошко его значение пересчитывается в окошках других типов матриц. Универсальным способом ввода является использование движка на биполярной шкале. Ввод единицы и нуля в матрицу фактов предпочтений осуществляется отклонением движка вправо/влево от точки «Равноценны».

За исключением МПС «Выигрыши/Потери», остальные типы матриц симметричны относительно главной диагонали. Это позволяет заполнять клетки только одной треугольной подматрицы. Значения $(N(N-1)/2)$ клеток симметричной треугольной подматрицы вычисляются на основе свойства симметрии.



Рис. 2. Вкладка «Предпочтения» СВП

При формировании МПС большой размерности эксперты испытывают трудности в нахождении предпочтений для всех пар сущностей. Поскольку в таких случаях некоторые клетки матрицы оказываются пустыми, возникает проблема их заполнения. Она может решаться как формированием подсказок экспертам, так и путем автоматического вычисления предпочтений. Для реализации обоих способов требуется дополнительная информация. В качестве таковой логично использовать условие неухудшения согласованности предпочтений, присущих исходной матрице.

Наиболее очевидным является соблюдение условия порядковой согласованности МПС, заключающееся в обеспечении транзитивности предпочтений. Оно напрямую реализуется для

МПС фактов предпочтений в процессе построения замыкающей дуги a_{ik} относительно уже введенных предпочтений a_{ij} и a_{jk} . Элемент a_{ik} МПС служит подсказкой для ввода последующих предпочтений, что исключает возможную противоречивость предпочтений (возникновение циклов в графе).

Минимальное число предпочтений, необходимое для построения транзитивного замыкания графа, определяется гамильтоновой цепью, включающей все сопоставляемые сущности (вершины графа доминирования). Поскольку гамильтонова цепь связывает N вершин графа $N-1$ дугами, минимизация ввода предпочтений достигается путем выбора только тех пар, в которых одна из сущностей не встречалась в предыдущих парах. Например, семью сравнениями вместо $(N(N-1))/2=28$ можно ограничиться при формировании матрицы размерностью 8×8 .

Для матриц с количественными предпочтениями недостаточно построить транзитивное замыкание графа доминирования. Необходимо определить способ нахождения количественных оценок предпочтений. Он основан на нахождении таких чисел в пустых клетках матрицы, которые доставляют максимум функции согласованности. Максимум достигается при $\lambda_{max} \rightarrow n$, где λ_{max} – максимальное собственное число матрицы, n – ее размерность [3].

Анализ свойств МПС выполняется на вкладке «Результат» (рис. 3).

Поскольку матрица на рисунке 2 построена на основе вектора приоритетов (рис. 1), она полностью согласована, что указывает на отсутствие циклов в графе предпочтений, полученном для этой матрицы (см. рис. 3). При необходимости выводится граф предпочтений. Его дуги в отсутствие циклов окрашены в черный цвет. С увеличением числа проходящих через них циклов предпочтений они розовеют. При наведении мыши на

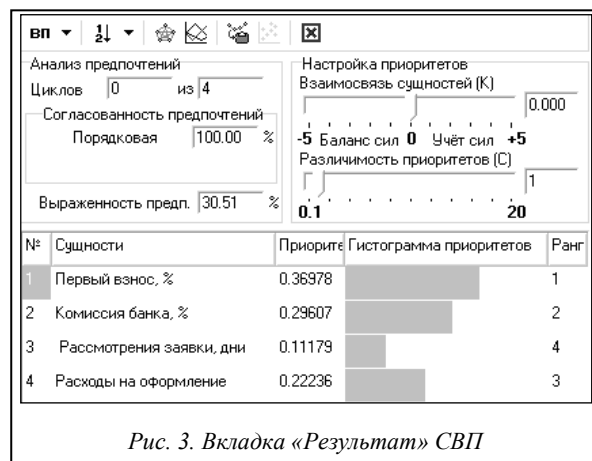


Рис. 3. Вкладка «Результат» СВП

дугу приводится список проходящих через нее циклов.

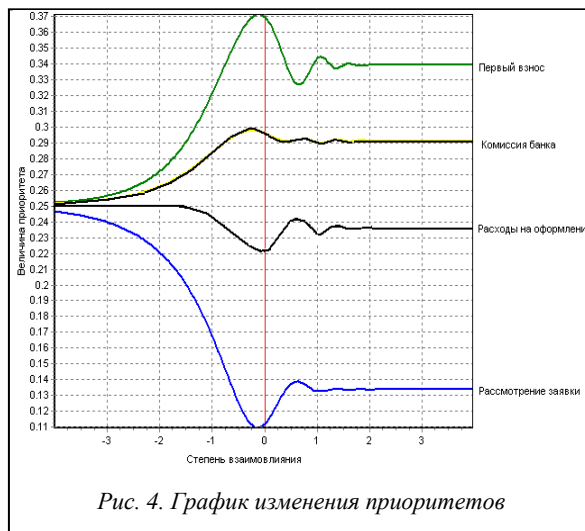
Показатель выраженности предпочтений отражает степень несимметричности элементов относительно главной диагонали. Для матрицы кратности предпочтений вычисляется коэффициент количественной (кардинальной) согласованности по формуле, предложенной Т. Саати.

В правой части вкладки «Результат» задаются параметры вычисления приоритетов сущностей и результаты вычислений, представленные в абсолютной и порядковой шкалах. Приоритеты сущностей вычисляются на основе матрицы A по итерационной формуле: $w=(c \cdot A + E)^k \cdot e^T$, где E – единичная матрица с элементами $e_{ii}=1$; e^T – транспонированный вектор текущих приоритетов с единичными начальными значениями.

Регулируемые параметры c и k представляют собой масштабный коэффициент ($c \geq 1$) и степень матрицы. Увеличение масштабного коэффициента улучшает различимость приоритетов, а при $k \rightarrow \infty$ величина приоритетов учитывает результаты взаимодействия сущностей между собой, то есть их силу. Изменение приоритетов сущностей в зависимости от значения степени k , измеренной в логарифмической шкале, иллюстрируется графиком (рис. 4).

СВП может использоваться для многокритериальной оптимизации по методу Саати. Участвующие в оптимизации МПС формируются как экспертами, так и автоматически, по значениям признаков из таблицы системы СВРЬ-Р. Формирование МПС на основании столбца таблицы осуществляется попарным сравнением его значений. В этом режиме из системы СВРЬ-Р задаются не только список сущностей и тип МПС для каждого признака, но и параметры вычисления приоритетов c и k . Указывается также способ формирования МПС.

Многокритериальная оценка приоритета сущностей вычисляется перемножением матрицы



приоритетов сущностей по каждому из критериев на вектор важности этих критериев. Он задается из системы СВРЬ-Р либо вычисляется на основе экспертных предпочтений в системе СВП. МАИ реализуется путем вычисления приоритетов в таблицах иерархии от листовых до корневой.

Результаты однокритериальной и многокритериальной оптимизации возвращаются в таблицы «Решение» системы СВРЬ-Р. Содержимое МПС с оценками их согласованности и выраженности предпочтений, а также приоритеты сущностей экспортируются в табличный процессор *MS Excel* для документирования и последующей обработки.

Литература

1. Саати Т.Л. Принятие решений при зависимостях и обратных связях. М.: Изд-во ЛКИ, 2007. 357 с.
2. Микони С.В. Многокритериальный выбор на конечном множестве альтернатив: учеб. пособие. СПб: Лань, 2009. 273 с.
3. Микони С.В., Киселев И.С. Приближенный метод определения матрицы парных сравнений с кратными предпочтениями // IEEE AIS'07; CAD-2007: тр. конф. (Дивноморское, 3–9 сентября 2007 г.). М.: Наука. Физматлит, 2007. Т. 1. С. 330–335.

ОНТОЛОГИИ ДЛЯ РЕАЛИЗАЦИИ ОБРАТНОЙ ТРАССИРОВКИ ПРИ РАЗРАБОТКЕ И СОПРОВОЖДЕНИИ ПРОГРАММ

В.В. Рогальчук; А.Д. Хомоненко, д.т.н.

(Петербургский государственный университет путей сообщения, joawl@mail.ru, khomon@mail.ru)

Дается краткая характеристика метода обратной трассировки кода, направленного на улучшение понимаемости программ в процессе их разработки и сопровождения. Обосновывается применение онтологий как средства представления знаний в процессе разработки и сопровождения программ. Предлагается вариант онтологии, предназначенной для представления знаний в процессе отслеживания и устранения ошибок программного проекта.

Ключевые слова: обратная трассировка программ, онтологии, представление знаний, разработка программ, сопровождение программ, программное обеспечение, ошибки, отношения.

На эффективность процессов разработки и сопровождения программного обеспечения (ПО)

большое влияние оказывает используемая технология. Одним из подходов, способствующих по-

вышению эффективности разработки и сопровождения ПО, является метод обратной трассировки [1]. Названный метод заключается в предоставлении разработчику инструментария, обеспечивающего поиск и навигацию от элементов программного кода к связанным с его созданием и изменением в ходе развития проекта документам. Это способствует лучшему пониманию и ускорению изучения кода программ. Так как задача изучения программ является одной из самых трудоемких в процессах разработки и сопровождения [2], использование обратной трассировки позволяет снизить общую стоимость проекта разработки. Наличие дополнительного инструментария увеличивает эффективность поиска и устранения ошибок, за счет чего повышается надежность программной системы.

При реализации метода обратной трассировки кода программ требуется решить задачу представления знаний о разрабатываемом проекте ПО. Одним из наиболее перспективных подходов к представлению знаний, по мнению многочисленных специалистов, является использование онтологий [3].

Онтология – модель представления знаний, комбинирующая в себе свойства семантических сетей и логических моделей. Понимание термина «онтология» варьируется в зависимости от области применения, но среди множества различных определений чаще всего используется определение Грубера: онтология – точная спецификация концептуализации [3]. Под концептуализацией понимается обобщение понятий и информации, необходимых для описания объектов и решения задач в выбранной области знаний – свойства, отношения, аксиомы, утверждения, ограничения.

В [3] предложено рассматривать онтологию как базу знаний, которая может отчуждаться от разработчика и физически делиться между пользователями. Онтологии как средство представления знаний разрабатываются для большого количества предметных областей.

Систематическое описание онтологий для области программной инженерии приводится в [4]. В частности, авторы описывают онтологии для методологий разработки ПО, его сопровождения и измерения характеристик. Для сопровождения ПО предложена следующая группа подонтологий: описания элементов программной системы, используемых в созда-

нии компьютерных знаний, организационной структуры, процесса сопровождения и домена приложения. Как показывает проведенный анализ, данные онтологии не в полной мере поддерживают решение задачи обеспечения понимаемости кода при обратной трассировке программы. В состав реализующей обратную трассировку среды разработки должны входить средства интеграции с существующими системами управления задачами и отслеживания сообщений об ошибках. Следовательно, в состав онтологической системы необходимо включить подонтологию описания задач разработки и сопровождения ПО, а также подонтологию описания ошибок, обнаруживаемых в процессе инспектирования программного проекта. На рисунке 1 показаны основные элементы пяти подонтологий и некоторые из связей между ними.

Подонтология управления задачами для разработчиков создана для интеграции системы управления задачами с системой реализации обратной трассировки. Любые реализации систем управления задачами оперируют понятиями *Задача* и *Комментарий*. *Задача* обсуждается *Сотрудниками* организации-разработчика. В ходе обсуждения с *Задачей* связывается множество *Комментариев*, каждый из которых может устанавливать новое *Состояние задачи* (*Новая задача*, *Изучается*, *Отложена*, *Выполнена*, *Закрыта*, *Возобновлена* и т.д.). В составе онтологической системы каждый комментарий является *Документом*, к ко-

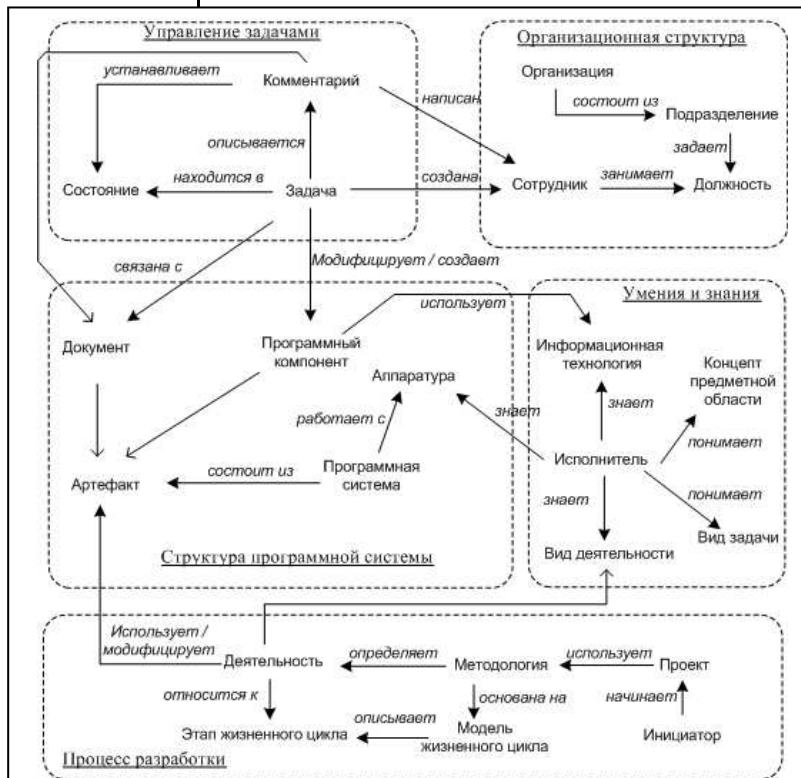


Рис. 1. Схема онтологии разработки и сопровождения ПО