

ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ

Государственное образовательное учреждение  
высшего профессионального образования

«САНКТ-ПЕТЕРБУРГСКАЯ ГОСУДАРСТВЕННАЯ ЛЕСОТЕХНИЧЕСКАЯ  
АКАДЕМИЯ имени С. М. Кирова»

---

*40 лет кафедре информатики и информационных систем*

ИНФОРМАЦИОННЫЕ СИСТЕМЫ  
И ТЕХНОЛОГИИ:  
ТЕОРИЯ И ПРАКТИКА

Сборник научных трудов

Санкт-Петербург  
2008

Рассмотрен и рекомендован к изданию  
Ученым советом факультета экономики и управления  
Санкт-Петербургской государственной лесотехнической академии  
10 июня 2008 г.

Редакционная коллегия:

**А. М. Заяц**, кандидат технических наук, профессор (отв. редактор),  
**И. В. Ганичев**, кандидат технических наук, доцент (отв. секретарь),  
**И. В. Панфилов**, доктор технических наук, профессор,  
**А. А. Михайлов**, кандидат технических наук, доцент

Рецензент  
доктор технических наук, профессор **В. И. Николаев**  
(Санкт-Петербургский филиал ОАО  
«Российские телекоммуникационные сети»)

УДК 630

**Информационные системы и технологии: теория и практика:**  
Сборник научных трудов. СПб.: СПбГЛТА, 2008. 200 с.

Юбилейный сборник научно-практических и методических работ сотрудников посвящен 40-летию юбилею одной из ведущих кафедр Санкт-Петербургской лесотехнической академии. Сборник подготовлен по материалам кафедры вуза, представленным на научно-технической конференции факультета экономики и управления ГЛТА в феврале 2008 года, и практических работ, выполненных ее сотрудниками.

ISBN 978-5-9239-0115-3

Темплан 2008 г. Изд. № 248.

© Санкт-Петербургская государственная  
лесотехническая академия (СПбГЛТА), 2008

## ОТ РЕДАКЦИИ

**Анатолий Михайлович Половко (1921–2007 г.г.)** – замечательный ученый и педагог, представитель славной плеяды советских – российских ученых, внесших значительный вклад в развитие науки и техники.

Творческий и научный потенциал Анатолия Михайловича Половко материализован в фундаментальных трудах по надежности вычислительной техники и информационных систем, а также по современным компьютерным технологиям. Только за период с 2002 по 2007 годы он написал и издал следующие работы:

- серию учебных пособий для студентов по использованию математических пакетов в решении инженерных задач (*Половко А. М. Derive для студента. Спб.: БХВ-Петербург, 2005.; Половко А. М., Ганичев И. В. Mathcad для студента. Спб.: БХВ-Петербург, 2006.*);

- двухтомную монографию, по собственному выражению Анатолия Михайловича, *«одно из главных дел его жизни»* (*Половко А. М., Гуров С. М. Основы теории надежности. 2-е изд., перераб. и доп. Спб.: БХВ-Петербург, 2006.; Половко А. М., Гуров С. М. Основы теории надежности: Практикум. Спб.: БХВ-Петербург, 2006.*).

Нам его друзьям, коллегам, ученикам и сослуживцам посчастливилось долгое время работать с Анатолием Михайловичем на кафедре, которой он отдал более тридцати лет своей жизни. Его профессиональная деятельность яркий пример беззаветного и бескорыстного служения НАУКЕ, которое продолжается его учениками и коллегами на кафедре *информатики и информационных систем*, а его научное наследие еще подлежит глубокому переосмыслению и изучению его учениками, последователями, учеными и специалистами.

В этом сборнике представлены статьи профессорско-преподавательского состава, отражающие основные результаты научных исследований, проведенных в 2007 году, посвященные сорокалетию кафедры *информатики и информационных систем* и светлой памяти заслуженного деятеля науки и техники, доктора технических наук, профессора Половко Анатолия Михайловича.

**А. М. Заяц**, кандидат технических наук, профессор

## **НОВЫЙ ПОДХОД В ПРАКТИЧЕСКОЙ ПОДГОТОВКЕ СТУДЕНТОВ ЛЕСОТЕХНИЧЕСКОЙ АКАДЕМИИ**

Опыт организации и ведения образовательного процесса в Лесотехнической академии показывает, что лесная отрасль с различными направлениями и видами лесного хозяйствования всегда требовала высококвалифицированных специалистов – практиков, обеспечивающих устойчивое управление лесами с целью неистощимого воспроизводства лесных ресурсов.

Задачи подготовки таких специалистов усложняются в обстановке стремительной разработки и внедрения новых технологий воспроизводства, заготовки и переработки древесины и в кардинально меняющихся условиях хозяйствования, определенных новым лесным кодексом.

В соответствии с государственными образовательными стандартами, практическая подготовка студентов лесотехнической академии, занимает от 12 до 16 недель, в зависимости от направления подготовки, причем в ходе обучения она начинается с учебной (ознакомительной) и заканчивается преддипломной практикой. В сферу практической подготовки также необходимо включить практические и лабораторные занятия по различным дисциплинам учебных планов.

Особенностью практической подготовки студентов лесотехнической академии, а на лесохозяйственном факультете в частности, является необходимость реализации ее задач на лесохозяйственных объектах, особенно для студентов старших курсов.

В академии имеется возможность проведения практик студентов на реальных лесохозяйственных объектах, таких как Лисинское и Охтинское лесничества, однако в современных условиях реформирования лесной отрасли, заключение договоров на проведение практик зачастую вызывает определенные трудности.

В связи с принятием нового лесного кодекса, проблема практической подготовки специалистов, профессиональный уровень которых должен обеспечить решение новых задач в условиях реформирования лесных хозяйств и всей отрасли в целом становится еще более актуальной.

Среди таких новых задач, предусмотренных кодексом, можно выделить следующие:

1. Разработка лесных планов субъектов Российской Федерации.
2. Разработка лесохозяйственных регламентов и проектов.
3. Проведение государственной экспертизы лесных деклараций, проектов освоения лесов, отчетов о воспроизводстве лесов и лесоразведении.
4. Разработка, формирование и сопровождение лесного реестра.
5. Обоснование и разработка полномочий органов местного самоуправления в области лесных отношений.
6. Применение информационных систем и технологий во всех областях лесного хозяйства.

Все эти задачи требуют от будущих специалистов лесной отрасли не только хороших теоретических знаний и практических навыков в предметной области, но и умений в сфере информационного обеспечения деятельности предприятия. Реализация последнего требования усложняется тем, что обучение в реальной информационной системе, развернутой на предприятии или в лесничестве практически невозможно.

Учитывая изложенные особенности необходимо искать новые подходы и формы в организации практической подготовки студентов.

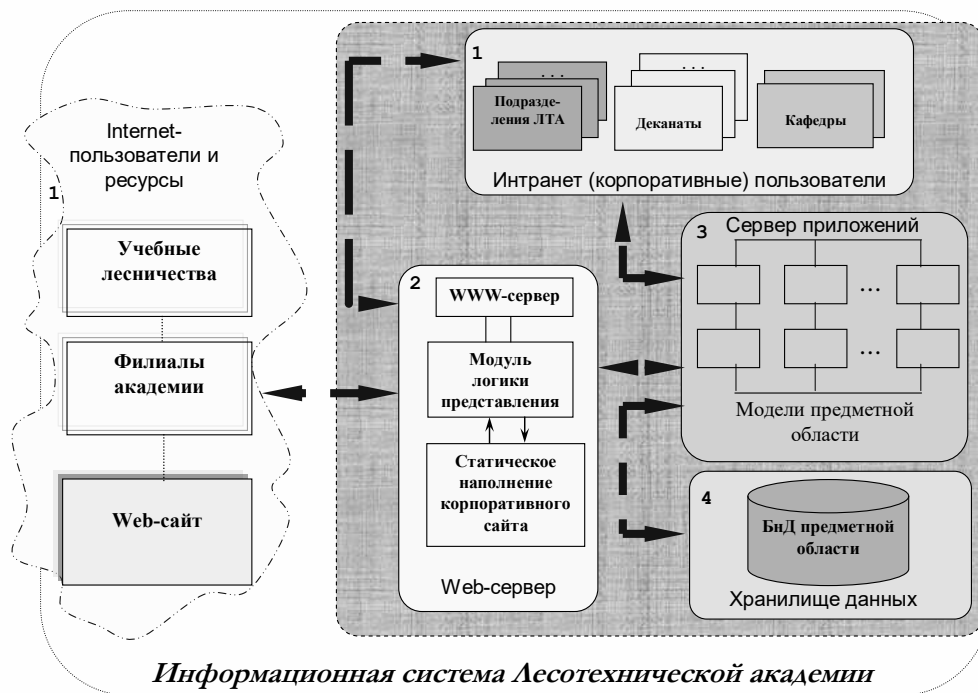
Одним из таких подходов является использование, в ходе практической подготовки студентов ГЛТА, новых информационных технологий реализуемых в учебных виртуальных лесничествах (УВЛ).

В контексте данной статьи виртуальные лесничества, как и виртуальные предприятия (ВП), представляют собой группы людей, совместно занимающихся общим делом, независимо от их физического местонахождения, работая в реальном времени (синхронно) или в отсроченном режиме (асинхронно), или ВП есть – сообщество географически разделенных работников, которые взаимодействуют в процессе производства, используя преимущественно средства телекоммуникаций.

Указанные цели виртуальных предприятий могут быть перенесены на задачи обеспечения качественной практической подготовки студентов в современной информационной среде в течение всего периода обучения.

Существо такого переноса заключается в использовании информационных ресурсов и технологий обработки данных внутри виртуальных лесничеств.

Образовательные функции виртуальных лесничеств определяются архитектурой их построения, эмулированной на аппаратно-программной платформе информационной системы ЛТА (см. рис. 1).



**Рис. 1.** Информационная система ГЛТА

С точки зрения учебных целей и задач, в составе УВЛ можно выделить следующие, относительно независимые компоненты: функциональную, учебно-методическую, сопровождения и поддержки.

В состав функциональной составляющей, то есть базовой компоненты УВЛ, как интегрированной информационной системы лесничества (ИИСЛ) входят (см. рис.2):

- геоинформационная система (ГИС);
- одна из систем ERP-класса;
- общие базы данных пространственной (картографической), тематической, атрибутивной и документальной информации, определенной лесным кодексом;
- корпоративная сеть академии – как ее транспортная инфраструктура, соединенная с локальными сетями подразделений виртуального лесничества и глобальной сетью Internet.

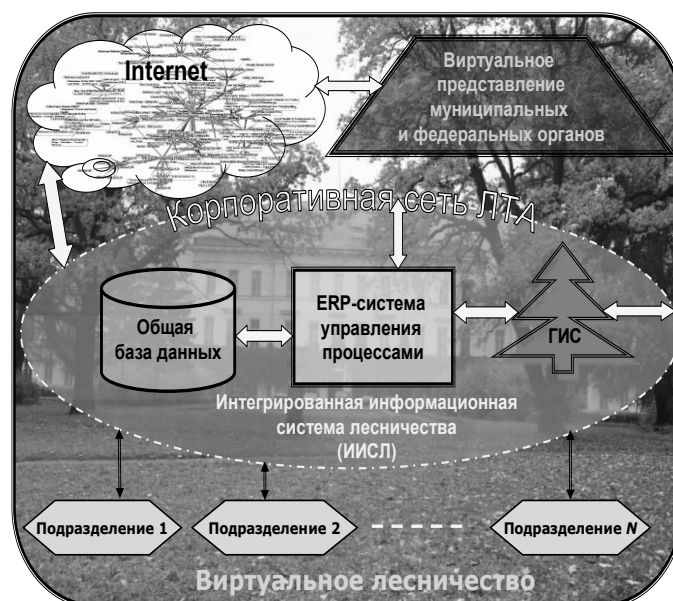


Рис. 2. Состав виртуального лесничества

Важным элементом обеспечивающим, необходимую технологию прохождения документов, является виртуальное представление муниципальных и федеральных органов, имеющих телекоммуникационную связь с виртуальным лесничеством посредством Internet/Intranet-технологий.

В архитектуре УВЛ используются многоуровневые клиент-серверные технологии. Количество серверов их назначение и номенклатура определяются функциональными задачами, решаемыми в интегрированной информационной системе лесничества.

Естественно, что для реализации ИИСЛ УВЛ потребуется соответствующая аппаратно-программная поддержка.

В качестве основных независимых сетевых служб в ней можно выделить:

- многопротокольную сетевую файловую службу;
- систему управления реляционной базой данных;
- единую систему авторизации доступа.

Каждая из названных подсистем должна поддерживаться отдельным сервером или группой серверов – кластером.

В качестве сетевой файловой системы можно использовать NFS ОС Solaris, которая является промышленным стандартом распределенной файловой системы, предоставляющей прозрачный доступ к удаленным файлам и каталогам, расположенным в произвольных узлах сети. Поскольку про-

токолы NFS пользуются широкой поддержкой, пользователи имеют доступ к файлам, располагающимся на ПК, рабочих станциях или АРМах, независимо от типов операционных систем, установленных на них. Операционная система Solaris поддерживает многоплатформность (IBM PC, Sun SPARC, IBM RS/6000), поэтому пользователи имеют возможность выбрать аппаратную основу, соответствующую их текущим нуждам и задачам.

Выбор данной ОС обуславливается ее универсальность, надежность, управляемость и доступность. При этом взаимодействие серверов между собой, а также с клиентскими компьютерами, имеющими сходные операционные системы, происходит при помощи стандартных средств UNIX. Обмен с рабочими станциями, работающими под управлением отличных от UNIX операционных систем, организуется при помощи виртуальных файловых серверов, поддерживаемых специальными серверными приложениями UNIX.

Система управления реляционной базой данных является наиболее перспективной в плане развития функциональности. Большинство приложений, создаваемых или приобретаемых для развития информационной системы академии, являются клиентами данной сетевой службы.

Единая система авторизации доступа призвана централизовать хранение учетной информации таким образом, чтобы при запросе пользователя по доступу к сетевым ресурсам серверы, поддерживающие запрошенные ресурсы, проверяли информацию об имени и пароле пользователя в единой базе данных

Операционные системы семейства UNIX, поддерживают самые различные способы идентификации пользователей благодаря технологии РАМ. Это позволяет серверам, работающим на платформе UNIX, авторизовать пользователей и через учетную базу домена MS Windows NT.

В рамках виртуального лесничества студенты могут решать все основные практические задачи, определенные государственными образовательными стандартами, такие как:

- выполнение документальных работ по всем лесохозяйственным направлениям, связанных с мониторингом, таксацией лесных массивов, организацией учета леса и созданием баз данных тематической и атрибутивной информации и т.п.;



- разработка и формирование документов, определенных лесным кодексом, таких как лесные планы и реестры, лесохозяйственные регламенты, проекты и др.;

- освоение технологий и процедур административного управления лесничеством, а также исполнения документов в рамках организационно-штатной структуры;

- оценка производственных и непроизводственных затрат на обеспечение требуемого уровня качества технологического процесса и экологических последствий;

- освоение современных информационных систем и технологий применительно к конкретной предметной области;

- выполнение научно-исследовательских работ и информационное обеспечение разработки курсовых и дипломных проектов.

Для успешной реализации этих задач необходимо иметь рассмотренную аппаратно-программную платформу, а также, что очень важно, пакет специального учебно-методического обеспечения поддержки (вторая компонента) виртуального лесничества и выполнения им функций в практическом обучении студентов. Эти методические материалы должны содержать в своем составе:

- описание структуры, состава и функций основных подразделений и лесничества в целом;

- состав, формы, технологии и процедуры, формирования, оформления и проводки основных документов лесохозяйственной деятельности;

- технологии проведения основных лесохозяйственных работ, таких как: таксация, разработка лесохозяйственных регламентов и лесных планов, проведение экспертизы лесных деклараций, разработка проектов освоения лесов, отчетов о воспроизводстве лесов и лесоразведении и т.п.;

- функциональные обязанности основных должностных лиц лесничества;

- методические указания по работе в интегрированной информационной системе лесничества;

- варианты заданий по практической подготовке и т.п.

В третью составляющую – компоненту сопровождения и поддержки включена подсистема регистрации студентов, комплексы рабочих прило-

жений студентов и преподавателей, средства, обеспечивающие автоматизированный доступ студентов к ресурсам и фондам фундаментальной библиотеки академии, а также другие механизмы обеспечения образовательного процесса «внутри» учебного виртуального лесничества.

Каждый студент в рамках предложенной архитектуры и назначенного (выбранного) варианта может создавать свое учебное виртуальное лесничество с учетом внутренних и внешних информационных и ресурсных связей, определенных заданием.

Основой создания своего УВЛ должна стать разработка студентом модели интегрированного информационного пространства на основе баз данных (БД) о подразделениях, их функциональных возможностях и технологических операциях.

Для уменьшения нагрузки на студенческие рабочие станции, работающие в режиме удаленных терминалов, и обеспечения более гибкой настройки режима распределенного выполнения функциональных процедур используется модель сервера приложений и web-сервера реализующего Internet/Intranet-технологии. При этом хранимые процедуры в каждом УВЛ, автоматически запускаются и выполняются по запросу с клиентских (студенческих) рабочих станций. Доступ к БД виртуальных лесничеств при их формировании, заполнении и использовании осуществляется как по корпоративной сети академии, так и через Интернет-сайт ЛТА ([www.ftacademy.ru](http://www.ftacademy.ru)).

Количество активных приложений, в открытых УВЛ, зависит от производительности серверов и пропускной способности корпоративной сети.

При организации информационного взаимодействия внутри интегрированной информационной системы лесничества должна обеспечиваться возможность использования данных различных форматов с их четкой классификацией на основе единых стандартов, используемых в ERP системах и ГИС.

Функциональное ядро ИИСЛ составляет система оперативного управления, связанная с производственно-экономической системой класса ERP, в которой актуализируются и обрабатываются данные по выполнению лесохозяйственных технологий и процессов. Вся необходимая информация хранится в общей базе данных, содержащей, как таксационную, так и информацию о подразделениях и об агентах – участниках ВЛ.

Так в академии, в качестве ERP системы, используется развернутая на кафедре информатики и информационных систем комплексная автоматизированная многопользовательская система управления предприятием «Галактика».

Функциональное ядро ИИСЛ отвечает за реализацию основных технологических операций документооборота и управление заказами клиентов, а также определение возможностей эффективной реализации их подразделениями.

Одной из главных задач функционального ядра в виртуальном лесничестве является обеспечение технологии приема заказа, проверку возможности его выполнения и распределение работ по его выполнению между отдельными подразделениями – участниками ВЛ. При этом программные механизмы, реализующие эти задачи должны позволять студентам получить практические навыки в использовании процедур управления в профессиональной деятельности.

Задачей систем оперативного управления подразделений является формирование оперативной информацией о процессах по отдельным операциям и снабжение этой информацией ИИСЛ.

В ходе реализации процедур оперативного управления лесохозяйственной деятельностью в рамках УВЛ студенты решают следующие практические задачи:

- проектирование и построение комплексных моделей процессов;
- привязка процессов к ресурсам, требующимся для их поддержки;
- привлечение ресурсов других компаний, когда это необходимо и целесообразно;
- выбор алгоритмов выполнения решения и оптимизации процессов;
- моделирование процессов на основе динамической модели – оперативного управления с использованием данных об ассортименте, технологии, загрузки мощностей, складских запасах, затратах, сроках.

Результатом работы системы оперативного управления является решение о возможности выполнения заказа с требуемыми параметрами на основе координации ресурсов в УВЛ.

Важно заметить, что в результате решения задач студенты каждый раз могут перестраивать свое УВЛ, что естественно недопустимо делать в информационных системах реальных лесничеств. Все учебные лесничества

могут храниться на сервере академии и студенты входят в свое учебное ВЛ под собственным логином и паролем и могут вносить изменения и производить настройки только в программной среде своего УВЛ, в соответствии с технологией решения поставленной задачи.

Предложенный подход, по использованию возможностей виртуальных лесничеств, ни в коей мере не отрицает традиционных методик проведения практик со студентами на реальных лесохозяйственных объектах, а он предполагает их разумное сочетание.

В рамках УВЛ могут оказываться и платные услуги Интернет-пользователям, которые хотят и могут использовать ресурсы виртуальных лесничеств для решения своих бизнес задач и практической подготовки сотрудников.

Использование современных информационных технологий и ресурсов в рамках учебных виртуальных лесничеств, позволит снизить временные и финансовые затраты и эффективно решить задачи практической профессиональной подготовки студентов в соответствии с требованиями государственных образовательных стандартов и запросами потенциальных работодателей.

Так на примерах, разработанных УВЛ легко продемонстрировать потенциальным работодателям имеющиеся у молодого специалиста навыки практической работы в предметной области с использованием современных информационных технологий, что обеспечивает востребованность и устойчивое положение выпускников лесотехнической академии на рынке труда.

**А. М. Заяц**, кандидат технических наук, профессор,  
**А. С. Гоголевский**, студент

## **АВТОМАТИЗАЦИЯ РАЗРАБОТКИ ЛЕСОХОЗЯЙСТВЕННЫХ РЕГЛАМЕНТОВ И ПРОЕКТОВ ОСВОЕНИЯ ЛЕСОВ НА ПЛАТФОРМЕ ВИРТУАЛЬНЫХ ЛЕСНИЧЕСТВ**

В соответствии с содержанием Части 1 Статьи 23 Лесного кодекса РФ основными территориальными единицами управления в области использования, охраны, защиты, воспроизводства лесов являются лесничества и лесопарки.

Основой осуществления использования, охраны, защиты, воспроизводства лесов, расположенных в границах лесничества, лесопарка, являются их лесохозяйственные регламенты (ЛХР). Без лесохозяйственного регламента нельзя разрабатывать проект освоения лесов, заключать договора аренды лесных участков и договора купли-продажи лесных насаждений, проводить уход за лесом, осуществлять его охрану и защиту.

Основными задачами, стоящими перед лесничествами являются:

- государственный контроль по реализации лесохозяйственных регламентов;
- организация аукционов по продаже права на заключение договоров аренды лесного участка или договоров купли-продажи лесных насаждений, если такое право ему делегировал орган исполнительной власти субъекта РФ;
- организация формирования лесных участков;
- сбор данных для Государственного лесного реестра, ведение отраслевой статистической отчетности и др.

Анализ контента «лесных» сайтов показывает, что на многих из них опубликованы проекты лесохозяйственных регламентов, проектов освоения лесов и другой документации, обеспечивающей жизнедеятельность лесной отрасли.

В опубликованных Web-документах лесохозяйственный регламент включает три раздела:

1. Общие сведения.

2. Нормативы, параметры и сроки разрешенного использования лесов, нормативы по охране, защите и воспроизводству лесов.

3. Ограничения использования лесов.

Все эти разделы содержат более двух десятков таблиц с различным количеством записей и полей.

В общих сведениях табличные данные сопровождаются схематическими картами субъекта РФ и территории лесничеств, для которых разрабатываются ЛХР, а также поквартальные карты-схемы подразделения лесов по целевому назначению. Каждый лесохозяйственный регламент сопровождается текстовой частью в виде пояснительной записки.

Во второй главе «Нормативы, параметры и сроки разрешенного использования лесов, нормативы по охране, защите и воспроизводству лесов» данные для заполнения таблиц распределения территорий по категориям земель, а также по расчетным лесосекам при различных видах рубок должны быть получены путем проведения расчетов.

Такой документ, как проект освоения лесов определен Статьей 88 Частью 2 Лесного кодекса и определяет порядок освоения лесов в целях обеспечения их многоцелевого, рационального, непрерывного, неистощительного использования, а также развития лесной промышленности на территории арендованных участков.

Проект освоения лесов является основным документом арендаторов – лесопользователей, которые должны представить его на государственную экспертизу и после утверждения, на его основании вести свою деятельность, и ежегодно заполнять лесные декларации.

Проект освоения лесов состоит из следующих разделов:

- 1) общие сведения;
- 2) сведения о лесном участке;
- 3) организация использования лесов;
- 4) создание и эксплуатация объектов лесной и лесоперерабатывающей инфраструктуры, а также объектов, не связанных с созданием лесной инфраструктуры;
- 5) мероприятия по охране, защите и воспроизводству лесов;
- 6) мероприятия по охране, использованию объектом животного мира, водных объектов.

В его состав входят более четырех десятков таблиц различной структуры и объема.

Трудность его разработки, как отмечают специалисты, состоит в том, что это достаточно сложная и дорогостоящая процедура. Монополии лесоустройства на разработку проектов освоения лесов нет, и арендатор может выполнить эту работу самостоятельно, но для него это будет очень сложно, так как сегодня нет той базы, на основе которых арендаторы должны разработать проекты освоения лесов и многие из них не являются специалистами лесной отрасли.

При этом на сегодняшний день отсутствует информация о методике расчетов и заполнения таблиц в рассмотренных документах, как в ручном, так и в автоматизированном виде.

Косвенно этот факт подтверждается тем, что в только конце 2007 года Фонд информации по лесным ресурсам (ФГУ «Лесоинфотека») объявил о проведении открытого конкурса по выбору подрядчиков работ в сфере информационно-коммуникационных технологий, обеспечивающих весь комплекс задач по автоматизированному созданию и сопровождению документов, определенных Лесным кодексом.

Данный проект реализуется на уровне Федерального агентства лесного хозяйства (Рослесхоза), однако основная работа по обеспечению реализации лесохозяйственных регламентов осуществляется в лесничествах, а проекты освоения лесов непосредственно реализуются арендаторами-лесопользователями. Лесохозяйственные регламенты разрабатываются на конкурсной основе любой организацией (и не только лесоустроительной), а утвержденные проекты освоения лесов могут зачастую разрабатываться и реализовываться неспециалистами.

Поэтому стоит задача создания довольно простой информационно-аналитической системы и методик, обеспечивающих создание и сопровождение лесохозяйственных регламентов, проектов освоения лесов, ежегодных деклараций и других документов, определенных Лесным кодексом и способствующих их качественной, своевременной и малозатратной разработке.

Одним из способов решение этой задачи может быть использование информационных ресурсов и технологий учебных виртуальных лесничеств

(УВЛ), эмулированных на аппаратно-программной платформе информационной системы лесотехнической академии (ЛТА).

Виртуальные лесничества, как и виртуальные предприятия (ВП), представляют собой группы людей, совместно занимающихся общим делом, независимо от их физического местонахождения, работая в реальном времени (синхронно) или в отсроченном режиме (асинхронно), или ВП есть – сообщество географически разделенных работников, которые взаимодействуют в процессе производства, используя преимущественно средства телекоммуникаций.

Указанные цели виртуальных предприятий могут быть перенесены на задачи разработки и ведения лесохозяйственных регламентов и документов их сопровождающих.

Существо такого переноса заключается в использовании информационных ресурсов и технологий обработки данных внутри виртуальных лесничеств (ВЛ).

В целях качественной разработки рассмотренных документов и обработки методики лесохозяйствования ресурсы аппаратно-программной платформы и реализуемые технологии виртуальных лесничеств должны обеспечивать реализацию общесистемных процедур и решение специальных прикладных задач, присущих лесному хозяйству.

К общесистемным процедурам следует отнести:

- авторизованный вход в систему и возможность групповой обработки документов с сохранение их образов;
- создание произвольного набора параметров для каждого вида документов и хранение в базе данных своего УВЛ как полного образа документа, так и только параметров, подставляемых в шаблон;
- применение открытых интерфейсов для настройки правил автоматического формирования содержания ЛХР и документов их сопровождающих с использованием информации из различных Интернет/Интранет ресурсов;
- ведение журнала наиболее важных операций над данными и запуска процедур импорта данных с возможностью их удаления в конкретном сеансе;
- назначения сценариев обработки на определенные действия (открытие, сохранение документов и т.п.);



- присвоение и изменение статуса документов, внутри своего виртуального лесничества, с сохранением даты его изменения и имени пользователя, его изменившего, поддержка и контроль версий одного и того же документа;
- автоматическое присвоение учетных и регистрационных номеров документов с гарантией отсутствия одинаковых в многопользовательском режиме и полнотекстовый поиск по содержанию документов внутри базы данных своего УВЛ;
- поддержка универсальной поисковой системы, предоставляющей возможность поиска, как по справочнику определенного типа, так и по всей справочной системе и работа со справочной и служебной информацией в привычном стиле Windows – «Проводник»;
- получение разработанных документов в различных форматах и выдача их на различные устройства вывода с возможностью публикации в Интернет.

В перечень специальных прикладных задач следует включить:

- проведение учета всех изменений, происходящих в лесном фонде, и отображение их в повыведельных базах данных и на лесных картах одновременно;
- обеспечение преемственности данных и разработанных прикладных программ пользователей в предыдущей версии или в других системах подобного типа;
- учет непрерывного обновления повыведельной и картографической баз данных о лесном фонде и формировании отчетных документов об изменениях в лесном фонде и лесохозяйственной деятельности органов лесного хозяйства;
- поддерживать совмещенные (атрибутивные и картографические) базы данных в актуальном состоянии;
- формирование плана рубок и расчет данных лесопользования;
- распределение площадей по категориям лесов и категориям земель, распределение площади и запасов насаждений по преобладающим породам и группам возраста;
- расчет объемов (площади и запасов) хозяйственных мероприятий;
- учет типа преобладающего ландшафта, типа леса, породы, группы возраста, класса бонитета;

- проектирование лесных дорог и объектов транспортной инфраструктуры;

- расчет среднего класса пожарной опасности и т.п.

Количество и номенклатура специальных прикладных задач может изменяться в зависимости от требований пользователей, размеров и разновидностей, реальных арендуемых участков и их таксационных характеристик.

Использование современных информационных технологий и ресурсов учебных виртуальных лесничеств, позволит пользователям снизить временные и финансовые затраты для разработки лесохозяйственных регламентов, проектов освоения лесов, ежегодных деклараций и других документов, определенных Лесным кодексом, отработать методику лесопользования и эффективно решать задачи по реализации требований этих документов в реальных лесничествах и лесопарках.

**В. А. Богатырев**, доктор технических наук, профессор,  
**С. В. Богатырев**, аспирант

## ВЕКТОРНАЯ ОПТИМИЗАЦИЯ СТРУКТУРЫ КЛАСТЕРА

**Введение.** Центры обработки данных современных корпоративных информационных системы, как правило, строятся на основе компьютерных сетевых технологий и включают в свой состав резервированные серверные системы различного функционального назначения, объединяемых в кластеры. При проектировании центров обработки данных должна решаться векторная задача оптимального выбора числа серверов различного функционального назначения, позволяющего при ограничении допустимых затрат на построение системы достичь максимум ее надежности и производительности. Задача решается с учетом потоков запросов, формируемых к различным серверам, и характеристик их надежности и производительности [1].

**Постановка задачи оптимизации.** В рассматриваемой компьютерной системе имеется  $N$  типов (по функциональному назначению) серверов. Серверы каждого типа объединяются в группы (кластеры), причем входящие в состав каждой группы серверы идентичны по параметрам.

Требуется найти распределения числа (кратность резервирования) серверов каждого типа  $(n_1, n_2, n_3, \dots, n_N)$ , обеспечивающее при заданных ограничениях на стоимость реализации системы ее максимальную надежность и производительность (минимальное время пребывания запросов в системе).

Исходными данными будем считать:

- вероятности (доли) распределения запросов на обслуживание в каждый из  $N$  типов групп (кластеров) серверов  $(b_1, b_2, b_3, \dots, b_N)$ , где  $\sum_{i=1}^N b_i = 1$ .
- средние времена выполнения запросов в серверах различного назначения (различных кластеров)  $(v_1, v_2, v_3, \dots, v_N)$ ;
- стоимости реализации серверов  $(c_1, c_2, c_3, \dots, c_N)$ ;
- суммарные средства, выделяемые на построение системы –  $c_0$ ;

▪ показатели надежности серверов (коэффициенты готовности) различных кластеров  $(p_1, p_2, p_3, \dots, p_N)$ ;

▪ средние времена выполнения запросов в системах хранения  $(u_1, u_2, u_3, \dots, u_N)$  и их показатели надежности  $(d_1, d_2, d_3, \dots, d_N)$ .

Будем считать, что система хранения данных непосредственно подключается к соответствующему серверному компьютеру.

Поставленная задача относится к векторной оптимизации. Решение задачи векторной оптимизации структуры кластера при заданной интенсивности суммарного потока запросов  $\lambda$  рассматривалась в [1]. В представленной работе задача векторной оптимизации структуры кластера решается при неопределенной интенсивности входного потока и неопределенности доли запросов к различным группам серверов (имеются предположения о вероятностях вариантов параметров входного потока).

В качестве критерия оптимизации воспользуемся критерием Байеса-Лапласа, представляемым в виде:

$$\text{Max}(\sum_{j=1}^h \pi_j A_j(n_1, n_2, \dots, n_N)),$$

где  $h$  и  $(\pi_1, \pi_2, \dots, \pi_h)$  число и вероятности возможных вариантов входного потока запросов (параметров входного потока)  $(\sum_{j=1}^h \pi_j = 1)$ , а  $(\lambda_1, \lambda_2, \dots, \lambda_h)$

интенсивности запросов для возможных вариантов входного потока;

$A_j(n_1, n_2, \dots, n_N)$  – показатель эффективности варианта структуры серверной системы, идентифицируемой числом серверов каждого типа, при  $j$ -м варианте параметров входного потока. Если варианты входного потока различаются как по интенсивности, так и по распределению вероятностей запросов к различным группам серверов, то вариантам входного потока сопоставляется матрица  $\|b_{ji}\|_{h \times N}$ ,  $j$ -я строка которой соответствует вероятностям (долям) того, что при поступлении запроса он предназначен для обслуживания в  $i$ -группе серверов (интенсивность суммарного потока запросов для  $j$ -го варианта равна  $\lambda_j$ ).

Сложность задания показателя эффективности обусловлена многокритериальностью рассматриваемой задачи. Зададим  $A_j(n_1, n_2, \dots, n_N)$  на основе: главного критерия – надежность, главного критерия – производительность, аддитивного критерия, мультипликативного критерия. В качестве комплексного показателя воспользуемся также коэффициентом сохранения эффективности [2, 3].

Будем считать заданными:  $(v_1, v_2, v_3, \dots, v_N)$ ,  $(u_1, u_2, u_3, \dots, u_N)$ ,  $(b_1, b_2, b_3, \dots, b_N)$ ,  $(p_1, p_2, p_3, \dots, p_N)$ ,  $(d_1, d_2, d_3, \dots, d_N)$   $\lambda$ ,  $(c_1, c_2, c_3, \dots, c_N)$ ,  $c_0$ .

Требуется найти такое распределение числа  $(n_1, n_2, n_3, \dots, n_N)$  серверов в каждом из  $N$  кластеров, при котором достигается максимум эффективности системы.

Рассмотрим следующие постановки оптимального проектирования кластерной системы серверов при использовании различных показателей  $A_j(n_1, n_2, \dots, n_N)$  эффективности вариантов структуры системы и представления целевой функции в форме критерия Байеса–Лапласа:

1) главный показатель – производительность:

$$\text{Max}\left(\sum_{j=1}^h \lambda_j \pi_j \frac{1}{T_j(n_1, n_2, n_3, \dots, n_N)}\right)$$

при условии  $\sum_{i=1}^N c_i n_i \leq c_0$ ,  $n_i \geq 1$  и  $P(n_1, n_2, n_3, \dots, n_N) \leq P_0$ ;

2) главный показатель – надежность:

$$\text{Max}\left(\sum_{j=1}^h \lambda_j \pi_j P_j(n_1, n_2, \dots, n_N)\right) \text{ при } \sum_{i=1}^N c_i n_i \leq c_0, n_i \geq 1 \text{ и } T_j(n_1, n_2, n_3, \dots, n_N) \leq T_0;$$

3) мультипликативный скалярный критерий:

$$\text{Max}\left(\sum_{j=1}^h \lambda_j \pi_j \frac{P_j(n_1, n_2, \dots, n_N)}{T_j(n_1, n_2, n_3, \dots, n_N)}\right), \text{ при } \sum_{i=1}^N c_i n_i \leq c_0, n_i \geq 1;$$

4) аддитивный скалярный критерий:

$$\text{Max}\left(\sum_{i=1}^h \lambda_i \pi_i \left\{ \alpha_1 \frac{P_j(n_1, n_2, \dots, n_N)}{P_0} - \alpha_2 \frac{T_j(n_1, n_2, n_3, \dots, n_N)}{T_0} \right\}\right)$$

при условии  $\sum_{i=1}^N c_i n_i \leq c_0$ ,  $n_i \geq 1$ ;

5) коэффициент сохранения эффективности [2, 3]:

$$\text{Max}\left(\sum_{j=1}^h \lambda_j \pi_j K_j(n_1, n_2, \dots, n_N)\right) \text{ при условии } \sum_{i=1}^N c_i n_i \leq c_0.$$

При этом  $P_0$  и  $T_0$  соответственно предельно допустимые надежность и среднее время пребывания запросов в системе  $T_j(n_1, n_2, n_3, \dots, n_N)$  среднее время пребывания запросов при интенсивности входного потока  $\lambda = \lambda_j$ ,  $P_j(n_1, n_2, \dots, n_N)$  вероятность работоспособного состояния системы и критерий сохранения эффективности  $K_j(n_1, n_2, \dots, n_N)$ , вычисляемые с учетом того, что в работоспособном состоянии, при  $\lambda = \lambda_j$  должно выполняться условие  $T_j(n_1, n_2, n_3, \dots, n_N) \leq T_0$ ;  $\alpha_1, \alpha_2$  – весовые коэффициенты, характеризующие относительную важность частных показателей,  $\alpha_1 + \alpha_2 = 1$ .

Оптимизация по приведенным критериям требует оценки вероятности работоспособного состояния системы серверов и среднего времени пребывания в ней запросов. Рассмотрим эти оценки.

**Оценка среднего времени пребывания запросов в системе.** Среднее время обслуживания запросов в системе серверов зависят от интенсивности входного потока  $\lambda$  и от доли поступления запросов в различные группы серверов. При равномерном распределении запросов в каждый сервер  $i$ -й группы ( $i$ -го кластера) поступает поток с интенсивностью  $\lambda b_i / n_i$ , создавая загрузку сервера равную  $v_i \lambda b_i / n_i$ , при этом для  $j$ -го варианта входного потока  $\lambda = \lambda_j$ , если распределение доли запросов к различным группам серверов меняется и задано матрицей  $\|b_{ji}\|_{h \times N}$ , то  $b_i = b_{ji}$ .

Каждый сервер и его систему хранения представим в виде системы массового обслуживания типа М/М/1 с неограниченной очередью. Будем

предполагать, что каждый запрос к серверу порождает одно обращение к системе хранения этого же сервера, тогда среднее время пребывания запросов в системе:

$$T(n_1, n_2, \dots, n_N) = \sum_{i=1}^N b_i \left( \frac{v_i}{1 - \frac{v_i \lambda b_i}{n_i}} + \frac{u_i}{1 - \frac{u_i \lambda b_i}{n_i}} \right),$$

причем для всех  $i$  должно соблюдаться условие  $\left(1 - \frac{v_i \lambda b_i}{n_i}\right) < 1$ ,  $\left(1 - \frac{u_i \lambda b_i}{n_i}\right) < 1$ , стационарности режима обслуживания.

**Оценка надежности системы.** Сервер отказывает, если неисправны его компьютер или система хранения. Будем считать, что исследуемая компьютерная система работоспособна, если исправен хотя бы один сервер в каждой группе (кластере). В этом случае вероятность работоспособного состояния системы (коэффициент готовности) определяется по формуле

$$P_c(n_1, n_2, n_3, \dots, n_N) = \prod_{i=1}^N (1 - (1 - p_i d_i)^{n_i}).$$

Приведенная оценка надежности не учитывает, что некоторые состояния, могут не удовлетворять требованиям по допустимому среднему времени пребывания запросов в системе и по вероятности превышения времени обслуживания некоторое пороговое значение.

Надежность серверной системы с учетом ограничений на максимальное среднее время пребывания запросов всех типов определим по формуле

$$P(n_1, n_2, n_3, \dots, n_N) = \sum_{k_1=1}^{n_1} \sum_{k_2=1}^{n_2} \sum_{k_3=1}^{n_3} \dots \sum_{k_N=1}^{n_N} \delta(k_1, k_2, k_3, \dots, k_N) C_{n_1}^{k_1} C_{n_2}^{k_2} C_{n_3}^{k_3} \dots C_{n_N}^{k_N} \times \\ \times p_1^{k_1} p_2^{k_2} p_3^{k_3} \dots p_N^{k_N} (1 - p_1)^{n_1 - k_1} (1 - p_2)^{n_2 - k_2} (1 - p_3)^{n_3 - k_3} \dots (1 - p_N)^{n_N - k_N},$$

где  $\delta(k_1, k_2, k_3, \dots, k_N)$  – условие работоспособности состояния серверной системы, зависящее от интенсивности входного потока  $\lambda = \lambda_j$ ,

$$\delta(k_1, k_2, k_3, \dots, k_N) = \begin{cases} 1, & \text{if } T(k_1, k_2, k_3, \dots, k_N) \leq T_0, \\ 0, & \text{if } T(k_1, k_2, k_3, \dots, k_N) > T_0, \end{cases} \quad (1)$$

причем  $T(k_1, k_2, k_3, \dots, k_N)$  – среднее время пребывания запросов в системе при исправности  $k_1, k_2, k_3, \dots, k_N$  серверов соответствующих кластеров,  $T_0$  – предельно допустимое среднее время пребывания запросов ко всем типам серверов. Условие работоспособности состояния системы  $\delta(k_1, k_2, k_3, \dots, k_N)$  серверов может также быть выражено через вероятность не превышения времени реакции системы (пребывания запросов в системе) заданное предельно допустимое время [4].

Надежность серверной системы с учетом дифференциации ограничений на максимальное среднее время пребывания запросов для каждой группы серверов (в каждом кластере) определим [1] как

$$P(n_1, n_2, n_3, \dots, n_N) = \prod_{i=1}^N \left( \sum_{k=1}^{n_i} \delta_{ik} C_{n_i}^k p_i^k (1-p_i)^{n_i-k} \right),$$

$$\delta_{ik} = \begin{cases} 1, & \text{if } T_{ik} \leq T_{0i}, \\ 0, & \text{if } T_{ik} > T_{0i}, \end{cases} \quad T_{ik} = \left( \frac{v_i}{1 - \frac{v_i \lambda b_i}{k}} + \frac{u_i}{1 - \frac{u_i \lambda b_i}{k}} \right),$$

$T_{0i}$  – максимально допустимое среднее время пребывания запросов для  $i$ -й группы серверов ( $i$ -го кластера),  $T_{ik}$  – среднее время пребывания запросов на обслуживание к  $i$ -й группе серверов и исправности среди них  $k$  серверов, зависящее от интенсивности входного потока  $\lambda = \lambda_j$ .

**Оценка коэффициента сохранения эффективности.** Коэффициент сохранения эффективности, применим как показатель надежности сложных систем, характеризуемых множеством состояний с постепенной потерей эффективности (производительности) по мере накопления отказов. Рассматривая в качестве показателя эффективности величину обратную среднему времени пребывания запросов, коэффициент сохранения эффективности оценим как



$$K(n_1, n_2, n_3, \dots, n_N) = \sum_{k_1=1}^{n_1} \sum_{k_2=1}^{n_2} \sum_{k_3=1}^{n_3} \dots \sum_{k_N=1}^{n_N} \frac{T(n_1, n_2, n_3, \dots, n_N)}{T(k_1, k_2, k_3, \dots, k_N)} C_{n_1}^{k_1} C_{n_2}^{k_2} C_{n_3}^{k_3} \dots C_{n_N}^{k_N} \times \\ \times p_1^{k_1} p_2^{k_2} p_3^{k_3} \dots p_N^{k_N} (1-p_1)^{n_1-k_1} (1-p_2)^{n_2-k_2} (1-p_3)^{n_3-k_3} \dots (1-p_N)^{n_N-k_N},$$

где  $T(n_1, n_2, n_3, \dots, n_N)$  и  $T(k_1, k_2, k_3, \dots, k_N)$  соответственно среднее время пребывания запросов системе в исходном состоянии (когда все  $(n_1, n_2, n_3, \dots, n_N)$  серверов,  $N$  кластеров исправны) и в состоянии, для которого число исправных серверов в  $N$  кластерах представлено вектором  $(k_1, k_2, k_3, \dots, k_N)$  [1].

Коэффициент сохранения эффективности при ограничении максимально допустимого среднего время пребывания запросов вычисляется как

$$K(n_1, n_2, n_3, \dots, n_N) = \sum_{k_1=1}^{n_1} \sum_{k_2=1}^{n_2} \sum_{k_3=1}^{n_3} \dots \sum_{k_N=1}^{n_N} \delta(k_1, k_2, k_3, \dots, k_N) \frac{T(n_1, n_2, n_3, \dots, n_N)}{T(k_1, k_2, k_3, \dots, k_N)} \times \\ \times C_{n_1}^{k_1} C_{n_2}^{k_2} C_{n_3}^{k_3} \dots C_{n_N}^{k_N} p_1^{k_1} p_2^{k_2} p_3^{k_3} \dots p_N^{k_N} (1-p_1)^{n_1-k_1} (1-p_2)^{n_2-k_2} (1-p_3)^{n_3-k_3} \dots (1-p_N)^{n_N-k_N},$$

где  $\delta(k_1, k_2, k_3, \dots, k_N)$  определяется по формуле (1).

Если для каждой группы серверов заданы ограничения на максимально допустимое среднее время пребывания запросов, то [1]:

$$\delta(k_1, k_2, k_3, \dots, k_N) = \prod_{i=1}^N \delta(k_i),$$

$$\text{где } \delta(k_i) = \begin{cases} 1, & \text{if } T_i(k_i) \leq T_{0i}, \\ 0, & \text{if } T_i(k_i) > T_{0i}, \end{cases} \text{ при этом } T_i(k_i) = \left( \frac{v_i}{1 - \frac{v_i \lambda b_i}{k_i}} + \frac{u_i}{1 - \frac{u_i \lambda b_i}{k_i}} \right).$$

Пример оптимизации

Расчеты проведем при [1]:

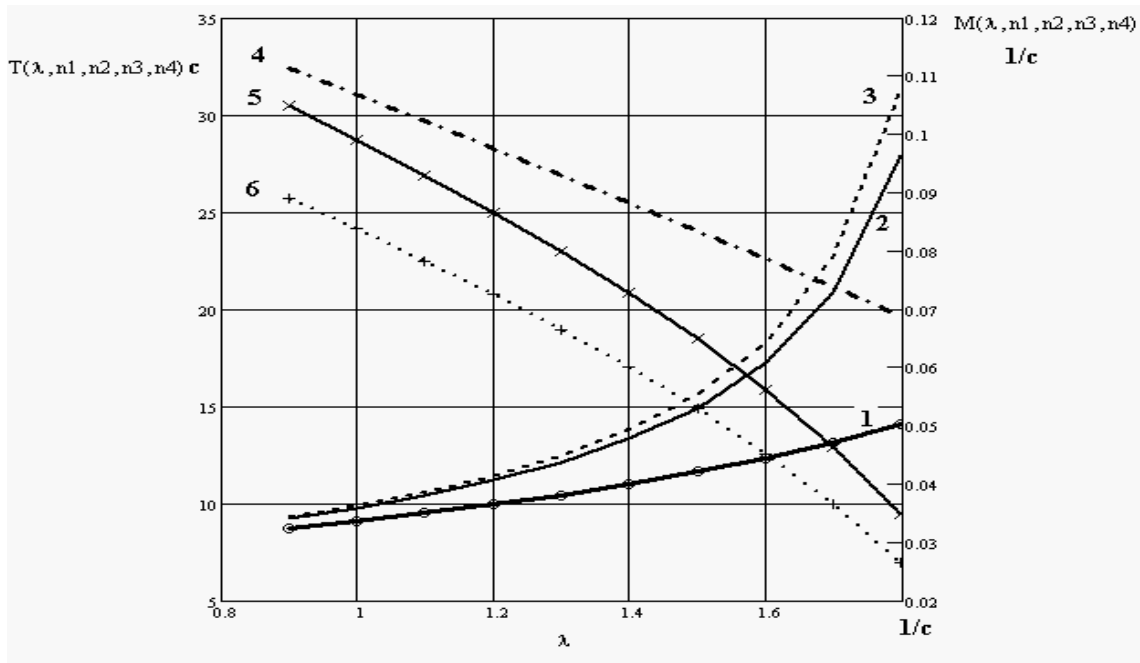
$$v = (1, 2, 5, 6) \text{ с.}, \quad u = (0, 1, 1, 0, 5, 0, 7) \text{ с.}, \quad b = (0, 1, 0, 2, 0, 5, 0, 2),$$

$$c = (10, 10, 10, 10) \text{ тыс. у.е.}, \quad c_0 = 200 \text{ тыс. у.е.},$$

$$N=4, p=(0.9, 0.9, 0.9, 0.9), d=(0.92, 0.92, 0.92, 0.92).$$

В результате оптимизации установлено, что для указанных выше параметров при интенсивности запросов  $\lambda \leq 0,8$  ( $1/c$ ) распределение числа серверов в группах (кластерах), обеспечивающее максимум мультипликативного критерия  $n=(3,6,7,4)$  шт., при интенсивности запросов  $0,8 < \lambda \leq 2,3$  ( $1/c$ ) – это распределение –  $n=(2,6,8,4)$  шт., а при  $2,4 < \lambda \leq 3$  ( $1/c$ ) –  $n=(1,7,8,4)$  шт.

На рис. 1 представлена зависимость среднего времени пребывания запросов в системе  $T(n_1, n_2, n_3, \dots, n_N)$  от интенсивности потока запросов  $\lambda$ . Кривая 1 соответствует случаю оптимального числа серверов в группах  $n=(2,6,8,4)$  шт.; кривая-2 числу серверов в группах  $n=(2,4,10,4)$  шт. пропорциональному вероятности запросов  $(b_1, b_2, b_3, \dots, b_N)$ ; кривая – 3 соответствует случаю  $n=(1,9,5,5)$  шт., когда число серверов в группах пропорционально времени  $(v_1, v_2, v_3, \dots, v_N)$  обработки в них запросов. Кривыми 4 – 6 представлены зависимости мультипликативного критерия  $M(n_1, n_2, n_3, \dots, n_N)$  от интенсивности запросов  $\lambda$  при оптимальном числе серверов в группах  $n=(2,6,8,4)$  шт. и числе серверов в группах  $n=(2,4,10,4)$  шт. и  $n=(1,9,5,5)$  шт.



**Рис. 1.** Зависимость времени пребывания запросов  $T(n_1, n_2, n_3, \dots, n_N)$  мультипликативного критерия от интенсивности  $\lambda$  запросов

Представленные зависимости подтверждают целесообразность проведения оптимизации серверных систем, причем эффект от оптимизации увеличивается с ростом загрузки.

Таким образом, поставлена задача оптимизации структуры системы, содержащей несколько групп резервированных серверов (кластеров) различного функционального назначения при различных вариантах, отличающихся интенсивностями суммарного входного потока запросов и долями запросов к различным группам серверов. В результате оптимизации на основе критерия Байеса-Лапласа по различным показателям эффективности (аддитивная и мультипликативная свертки, коэффициент сохранения эффективности и др.) определяется распределение числа серверов каждого типа, обеспечивающее наиболее рациональную структуру кластерной системы. Решение ищется при компромиссе достижения наибольшей надежности и наименьшего среднего времени ожидания запросов в системе при заданных ограничениях на стоимость реализации системы.

#### **Библиографический список**

1. *Богатырев В. А.* Оптимальное резервирование системы разнородных серверов / Приборы и системы. Управление, контроль, диагностика. № 12. 2007. С 30–35.
3. ГОСТ 27.002-89. Надежность в технике. Основные понятия. Термины и определения.
4. ГОСТ 27.003-90. Надежность в технике. Состав и общие правила задания требований по надежности.
5. *Богатырев В. А.* Оценка надежности и оптимальное резервирование кластерных компьютерных систем / Приборы и системы. Управление, контроль, диагностика. № 10. 2006. С. 18–21.

## ЭФФЕКТИВНОСТЬ ФУНКЦИОНИРОВАНИЯ ДИНАМИЧЕСКОЙ СИСТЕМЫ С ДИСКРЕТНЫМ МНОЖЕСТВОМ СОСТОЯНИЙ

Любая созданная человеком система должна обеспечивать некоторый эффект (выигрыш) от своего функционирования, то есть должна быть полезной. Понятие ценности и пользы не имеют универсального характера, даже когда они отражают интересы больших групп людей. Индивидуальные представления о них из-за весьма различных мотивов и взглядов сильно различаются. Это различие может быть ограничено, если рассматривать полезность принимаемых решений в инженерной и хозяйственной деятельности. Однако и здесь остается возможность субъективной оценки полезности небольшими группами и отдельными людьми.

На полезность, в конечном счете, оказывают влияние и такие плохо оцениваемые свойства, как наглядность, удобство в эксплуатации и некоторые факторы, просто не поддающиеся учету. Пользу в этом случае трудно оценить, и ее приходится описывать только рядом желаемых свойств.

Следуя терминологии теории надежности [1], предположим, что множество  $E$  всех состояний системы представляется в виде объединения подмножества  $E_+$  работоспособных (благоприятных) и подмножества  $E_-$  отказов (неблагоприятных) состояний.

Если  $S_i$  является состоянием исправной работы системы, то обозначим через  $a_i$  стоимость продукции, производимой в единицу времени. Если состояние  $S_i$  классифицируется как состояние отказа системы, то через  $a_i$  можно обозначить штраф за недовыпуск продукции или затраты на проведение ремонта в единицу времени (в этом случае будем считать, что  $a_i < 0$ ).

Аналогично переходам из состояния в состояние также могут быть сопоставлены числа  $a_{i,j}$ , имеющие следующий смысл. Если переход из состояния  $S_i$  в состояние  $S_j$  вызван отказом некоторого элемента системы, то  $a_{i,j}$  характеризует потери или убытки, связанные с этим отказом (считаем, что  $a_{i,j} < 0$ ). Если переход из состояния  $S_i$  в состояние  $S_j$  вызван ремонтом

некоторого элемента системы, то качество системы улучшается, и  $a_{i,j}$  характеризует прибыль или доход, связанные с восстановлением.

Полезность системы во время ее эксплуатации может оцениваться величиной «выигрыша»  $W(t)$ , который приносит система в момент времени  $t$ . Общий выигрыш системы складывается из выигрыша пребывания системы в состояниях и из выигрыша, который получается вследствие мгновенных переходов из состояния в состояние.

Средний выигрыш от пребывания системы в состоянии  $S_i$ , очевидно, равен произведению  $a_i$  на среднее время пребывания системы в этом состоянии  $m_i(t)$ , то есть он равен  $a_i m_i(t)$ .

Средний выигрыш, обусловленный переходом из состояния  $S_i$  в состояние  $S_j$ , очевидно, равен произведению  $a_{i,j}$  на среднее число таких переходов  $M_{i,j}(t)$ , то есть он равен  $a_{i,j} M_{i,j}(t)$ . Общий средний выигрыш системы в момент  $t$  определяется тогда следующим образом

$$W(t) = \sum_{S_i \in E} a_i m_i(t) + \sum_{S_i, S_j \in E} a_{i,j} M_{i,j}(t). \quad (1)$$

Формула (1) представляет собой функцию полезности системы, учитывающую как время пребывания системы в каждом состоянии, так и число переходов из состояния в состояние.

В частности, если множество отказовых состояний пусто, то  $W(t)$  характеризует величину среднего дохода системы в момент времени  $t$ :

$$W(t) = \sum_{S_i \in E_+} a_i m_i(t). \quad (2)$$

Важным частным случаем среднего выигрыша  $W(t)$  является понятие среднего риска системы. Риск является неизбежным атрибутом эксплуатации техники и одним из важнейших показателей безопасности. Такой риск называется техногенным.

Риск имеет место не только при эксплуатации техники. Он присутствует при любых аномалиях: неблагоприятные условия среды (метеорологический риск), неблагоприятная экономическая ситуация (экономический риск) и т. д.

В общем случае риском называется возможность потерь вследствие внутренних аномалий в системе или аномалий среды [2]. В большинстве случаев риск оценивается денежными единицами, хотя могут быть случаи, когда потери носят такой характер, когда оценить их деньгами трудно или даже невозможно. Из определения следует, что в любом случае риск является случайной величиной. Случайный его характер объясняется случайностью величины потерь. При расчетах, как правило, используется математическое ожидание риска.

Формула техногенного риска получается, если все коэффициенты  $a_i$  и  $a_{i,j}$  положить равными нулю, за исключением коэффициентов  $a_{i,j}$ , соответствующих переходам из исправных в отказовые состояния, для которых  $S_i \in E_+, S_j \in E_-$

$$W(t) = \sum_{S_i \in E_+, S_j \in E_-} a_{i,j} M_{i,j}(t). \quad (3)$$

Ограничимся случаем системы с двумя состояниями: исправное  $S_0$  и отказовое  $S_1$ . Примерами дискретных систем могут быть энергетические системы, транспортные системы, системы передачи данных, банки и др. В таких системах можно выделить множество исправных и множество отказовых состояний. Объединяя состояния указанных множеств, будем иметь систему с двумя состояниями. Из (1) следует, что средний выигрыш равен

$$W(t) = a_0 m_0(t) + a_1 m_1(t) + a_{01} M_{01}(t) + a_{10} M_{10}(t). \quad (4)$$

Пусть время перехода из состояния  $S_0$  в состояние  $S_1$  случайное с плотностью распределения  $f(x)$ , время перехода из состояния  $S_1$  в состояние  $S_0$  также случайное с плотностью распределения  $g(x)$ . Тогда (см. [2]) среднее число переходов из состояния в состояние рассчитывается по формулам

$$M_{01}(t) = \int_0^t \sum_{k=0}^{\infty} f^{*(k+1)} * g^{*(k)}(x) dx, \quad (5)$$

$$M_{10}(t) = \int_0^t \sum_{k=1}^{\infty} f^{*(k)} * g^{*(k)}(x) dx, \quad (6)$$

а среднее время пребывания в состояниях – по формулам

$$m_0(t) = (1 + M_{10}) * \bar{F}(t), \quad (7)$$

$$m_1(t) = M_{01} * \bar{G}(t), \quad (8)$$

где  $\bar{F}(t) = \int_t^{\infty} f(x) dx$  – вероятность безотказной работы системы в течение

времени  $t$ ,  $\bar{G}(t) = \int_t^{\infty} g(x) dx$  – вероятность того, что за время  $t$  система не

будет восстановлена, \* – операция свертки.

Рассмотрим сначала случай, когда время перехода между состояниями  $S_0$  и  $S_1$  имеет показательное распределение. Интенсивность отказа системы может принимать одно из трех значений:  $\lambda = 0,005$  час<sup>-1</sup>,  $\lambda = 0,001$  час<sup>-1</sup>,  $\lambda = 0,0005$  час<sup>-1</sup>. Интенсивность восстановления системы  $\mu = 0,5$  час<sup>-1</sup>. Требуется вычислить средний выигрыш системы при следующих значениях экономических показателей:  $a_0 = 16$  ден. ед./час,  $a_1 = -5$  ден. ед./час,  $a_{01} = -7800$  ден. ед.,  $a_{10} = 100$  ден. ед.

Временные характеристики  $m_0(t)$ ,  $m_1(t)$ ,  $M_{01}(t)$ ,  $M_{10}(t)$  определяются по формулам (5) и (6). Как показано в [2], для показательных распределений справедливы следующие значения:

$$m_0 = \frac{\mu}{\mu + \lambda} t + \frac{\lambda}{(\mu + \lambda)^2} (1 - e^{-(\mu + \lambda)t}), \quad m_1 = \frac{\lambda}{\mu + \lambda} t - \frac{\lambda}{(\mu + \lambda)^2} (1 - e^{-(\mu + \lambda)t}),$$

$$M_{01} = \frac{\mu\lambda}{\mu + \lambda} t + \frac{\lambda^2}{(\mu + \lambda)^2} (1 - e^{-(\mu + \lambda)t}),$$

$$M_{10} = \frac{\mu\lambda}{\mu + \lambda} t - \frac{\mu\lambda}{(\mu + \lambda)^2} (1 - e^{-(\mu + \lambda)t})$$

Подставляя эти выражения в  $W(t)$ , получим:

$$W(t) = a_0 \left( \frac{\mu}{\mu + \lambda} t + \frac{\lambda}{(\mu + \lambda)^2} (1 - e^{-(\mu + \lambda)t}) \right) + a_1 \left( \frac{\lambda}{\mu + \lambda} t - \frac{\lambda}{(\mu + \lambda)^2} (1 - e^{-(\mu + \lambda)t}) \right) + a_{01} \left( \frac{\mu\lambda}{\mu + \lambda} t + \frac{\lambda^2}{(\mu + \lambda)^2} (1 - e^{-(\mu + \lambda)t}) \right) + a_{10} \left( \frac{\mu\lambda}{\mu + \lambda} t - \frac{\mu\lambda}{(\mu + \lambda)^2} (1 - e^{-(\mu + \lambda)t}) \right).$$

Результаты расчетов приведены в табл. 1.

Таблица 1

t, час	Выигрыш системы, ден. ед.		
	$\lambda, \text{ час}^{-1}$		
	0,005	0,001	0,0005
0	0	0	0
10	-225	83	121
20	-448	165	243
30	-671	248	364
40	-894	331	485
50	-1118	414	607
60	-1341	496	728
70	-1564	579	849
80	-1787	662	971
90	-2011	744	1092
100	-2234	827	1213

Изменение функции выигрыша  $W(t)$  во времени для трех значений интенсивностей отказов приведено на рис. 1.

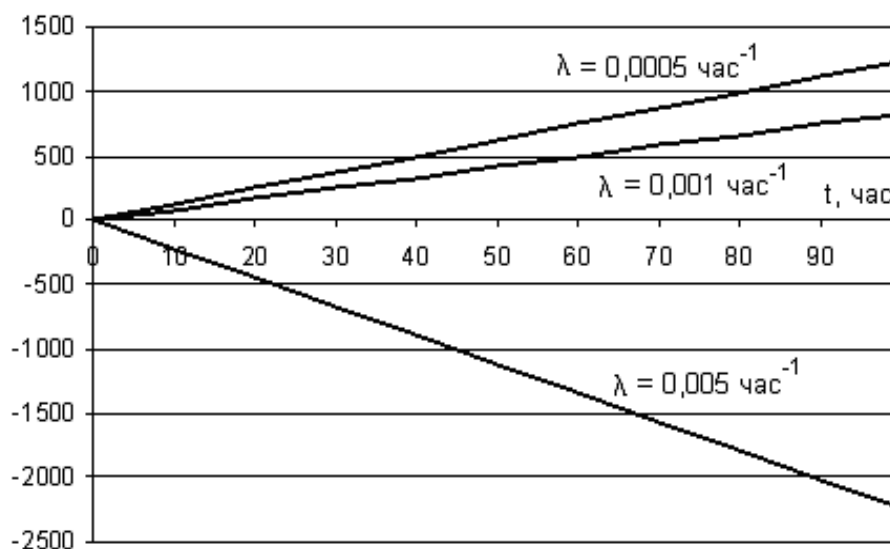


Рис. 1. Функции выигрыша для различной надежности системы



Относительный выигрыш системы является стабильным. Асимптотическое значение среднего выигрыша равно

$$W(t) = \frac{a_0\mu + a_1\lambda + (a_{01} + a_{10})\mu\lambda}{\mu + \lambda} t.$$

Для нашего случая относительный выигрыш составляет  $\frac{W(t)}{t} = -22,34$  при  $\lambda = 0,005 \text{ час}^{-1}$ ,  $\frac{W(t)}{t} = 8,27$  при  $\lambda = 0,001 \text{ час}^{-1}$ ,  $\frac{W(t)}{t} = 12,13$  при  $\lambda = 0,0005 \text{ час}^{-1}$ .

По результатам расчетов можно сделать следующие выводы:

- повышение надежности системы (уменьшение  $\lambda$ ) приводит к повышению выигрыша, при недостаточно высокой надежности техники выигрыш становится отрицательным; в нашем примере существенный проигрыш имеет место при  $\lambda = 0,005 \text{ час}^{-1}$ ;

- общий выигрыш системы растет с течением времени практически линейно (рис. 1);

- цена выпускаемой продукции из-за отказов техники может существенно снижаться, например, при отсутствии отказов стоимость выпускаемой продукции в единицу времени равна 16 ден. ед. Однако фактически, в основном вследствие риска из-за отказа системы, стоимость продукции в единицу времени уменьшается и составляет всего  $\frac{W(t)}{t} = 8,27$  ден. ед., если интенсивность отказов системы  $\lambda = 0,001 \text{ час}^{-1}$ .

Рассмотрим другой пример. Сравним величину техногенного риска энергетической системы при различных вероятностных характеристиках ее функционирования. Предположим, что время до отказа системы имеет нормальное распределение с параметрами: математическое ожидание  $a = 1000$  час, среднее квадратическое отклонение  $s = 300$  час. Время восстановления имеет нормальное распределение с параметрами  $b = 10$  час,  $r = 3$  час. Величина потерь, обусловленная отказом системы, составляет  $a_{01} = 1$  млн. руб. Требуется определить функцию техногенного риска системы. Определить техногенный риск системы в случае, когда время до

отказа и время восстановления системы имеют показательные распределения с теми же средними значениями. Сравнить полученные результаты.

Из формулы (4) следует, что риск системы определяется из соотношения  $W(t) = a_{01}M_{01}(t)$ .

Поэтому в соответствии с формулой (5) достаточно вычислить среднее число переходов системы из работоспособного в отказовое состояние.

Как известно [1], сумма независимых случайных величин, имеющих нормальные распределения, также подчинена нормальному распределению. При этом математическое ожидание суммы равно сумме математических ожиданий слагаемых, а дисперсия суммы равна сумме дисперсий. Поэтому функция  $f^{*(k+1)} * g^{*(k)}(t)$  является плотностью нормального распределения с параметрами  $m_k = (k+1)a + kb$  и  $\sigma_k = \sqrt{(k+1)s^2 + kr^2}$ . Отсюда следует, что

$$M_{01}(t) = \sum_{k=0}^{\infty} \Phi\left(\frac{t - m_k}{\sigma_k}\right), \quad (9)$$

где  $\Phi(t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^t e^{-\frac{x^2}{2}} dx$  – функция Лапласа с известным вычислительным алгоритмом.

Для показательных распределений среднее число переходов системы из работоспособного в отказовое состояние имеет вид

$$M_{01}(t) = \frac{\mu\lambda}{\mu + \lambda} t + \frac{\lambda^2}{(\mu + \lambda)^2} (1 - e^{-(\mu + \lambda)t}),$$

где  $\lambda = \frac{1}{a}$ ,  $\mu = \frac{1}{b}$ .

Проведенные расчеты (табл. 2) позволяют оценить выигрыш (риск) обеих систем в зависимости от законов распределения.

Таблица 2

t, час	Риск системы, млн. руб.		Относительная погрешность $\delta_w(t)$
	Нормальные распределения	Показательные распределения	
0	0,001	0	1
100	0,003	0,099	0,967
200	0,009	0,198	0,955
300	0,021	0,297	0,928
400	0,046	0,396	0,883
500	0,090	0,495	0,817
600	0,158	0,594	0,734
700	0,249	0,693	0,641
800	0,352	0,792	0,556
900	0,449	0,891	0,496
1000	0,519	0,990	0,476

На рис. 2 представлена зависимость от времени техногенного риска для системы с нормальными (кривая 1) и для системы с показательными (кривая 2) распределениями.

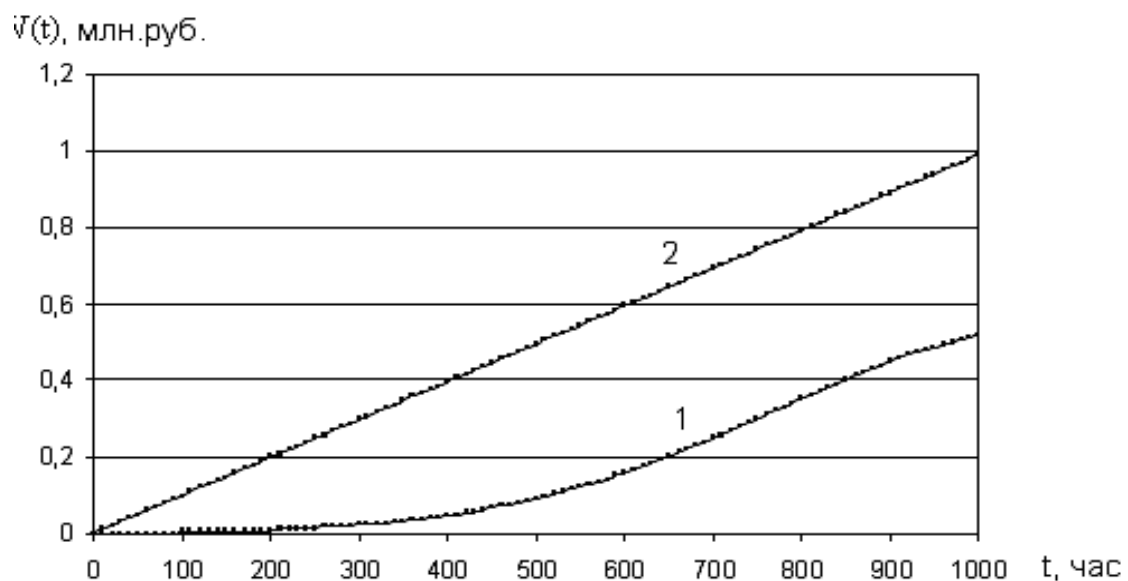
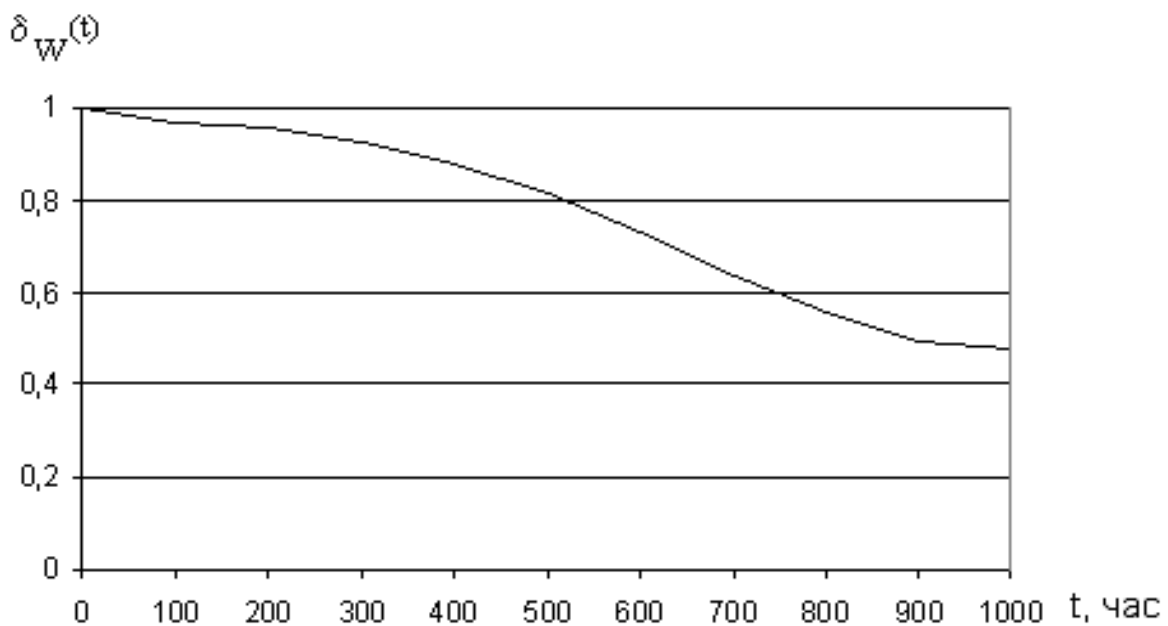


Рис. 2. Сравнение риска системы для различных распределений

Риск является возрастающей функцией времени. Риск существенно выше для показательных распределений особенно на начальном интервале функционирования системы (до 1000 час). Отсюда следует, что, оценивая

риск системы, важно иметь не только экономическую информацию о величине потерь от отказа системы, но также и вероятностные характеристики о ее отказах.

С течением времени зависимость риска от законов распределения ослабевает. В табл. 2 и на рис. 3 показано изменение относительной погрешности  $\delta_W(t) = \frac{W_n(t) - W_H(t)}{W_n(t)}$ , возникающей при замене нормальных распределений на показательные.



**Рис. 2.** Относительная погрешность в риске системы для различных распределений

Видим, что в начале эксплуатации системы различие в риске может достигать 100%, но с течением времени это различие убывает, и при  $t = 8000$  ч. погрешность составляет 5%.

### Библиографический список

1. Барзилович Е. Ю. и др. Вопросы математической теории надежности. / Под ред. Б. В. Гнеденко. М.: Радио и связь. 1983. 376 с.
2. Половко А. М., Гуров С. В. Надежность технических систем и техногенный риск. СПб.: Знание, 1998. 119 с.

**Л. В. Недорезов**, доктор физико-математических наук, профессор

## О НЕКОТОРЫХ КЛЮЧЕВЫХ ПРОБЛЕМАХ ЭКОЛОГИЧЕСКОГО МОДЕЛИРОВАНИЯ

**Введение.** В современном экологическом моделировании существует ряд весьма важных вопросов, которые являются не просто ключевыми, но решение которых может оказать существенное влияние на развитие данной науки в целом. В частности, одним из таких вопросов является следующий: что же именно описывают (динамику какой переменной) известные динамические модели Мальтуса и Ферхюльста [1–9], лежащие в основе практически всех современных моделей динамики популяций и экосистем? Как известно, модель Ферхюльста – это обобщение модели Мальтуса, связанное с учетом действия внутрипопуляционных саморегуляторных механизмов и имеющая вид

$$\frac{dx}{dt} = x(\alpha - \beta x). \quad (1)$$

Можно указать огромное число учебников и научных монографий (см., например, [1–10]), в которых сказано, что переменная  $x(t)$  – это численность популяции в некоторый момент времени  $t$ , параметр  $\alpha$  численно равен разности интенсивности рождаемости особей и их смертности, параметр  $\beta$  отражает влияние саморегуляторных механизмов на динамику численности (при  $\beta = 0$  модель (1) является моделью Мальтуса). Если после представления модели в данном виде сразу же перейти к вопросу о соответствии (в реальности, только визуального) поведения траекторий модели и многочисленных экспериментальных данных, то возникает иллюзия существования некоей супер-модели, которая одинаково хорошо описывает динамику самых разных популяций.

Крайне редко и неохотно авторы работ останавливаются на вопросе о допущениях, лежащих в основе модели (1). Поскольку ясно, что численность популяции является величиной целочисленной, то возникает вполне закономерный вопрос: что означает априорное предположение о диффе-

ренцируемости этой величины? Можно указать работы, в которых авторы пытаются устранить это очевидное несоответствие между моделью и объектом моделирования, но эти попытки ограничиваются исключительно математической стороной проблемы [6–10]. Что же означает это предположение с точки зрения биологии, вообще не рассматривается в литературе. Заметим, что предположение о том, что  $x(t)$  не численность, а плотность популяции (например, число особей на квадратный метр), также не спасает положения. И в этом случае, очевидно, переменная модели остается дискретной (принимаящей даже не счетное, а конечное множество значений).

Есть еще один вопрос, тесно связанный с первым. Если предположить, что модель (1) верна и описывает динамику численности популяции, то каким именно статистическим критерием следует пользоваться, чтобы корректно восстановить по экспериментальным данным значения параметров модели? Ясно, что этот вопрос относится не только к модели (1), а вообще ко всем математическим моделям популяционной динамики. Изобилие статистических критериев, используемых для нахождения параметров, не способствует улучшению ситуации или более глубокому пониманию сути проблемы. Вследствие этого возникает парадоксальная ситуация: с использованием различных критериев мы можем получить огромное количество различных оценок параметров модели. При этом до настоящего момента времени у нас нет каких-либо критериев для сравнения этих оценок между собой, что в конечном итоге не позволяет дать ответ на вопрос о том, какая же из полученных оценок наиболее близка к искомой.

Именно эти два вопроса рассматриваются в настоящей работе. Конечно, ключевых проблем в математической экологии намного больше (в частности, проблема выбора математического аппарата – в каких случаях следует предпочесть рекуррентные уравнения системам обыкновенных дифференциальных уравнений и почему, – проблема выбора вида функций, описывающих взаимодействие особей и так далее), однако, остановиться на каждой из этих проблем в одной публикации не представляется возможным.

**Как продифференцировать численность популяции?** Поскольку желание использовать обыкновенные дифференциальные уравнения для описания динамики численности биологических популяций появилось го-

раздо раньше, чем осознание того факта, что их использование встречает целый букет критических проблем [10–14], то в результате этого к настоящему моменту времени возникла весьма забавная ситуация: существует хорошо развитая теория, по сути, не имеющая никакой базы. Достаточно давно было понято [13, 15], что дифференцирование целочисленной переменной  $x(t)$  является незаконной операцией, соответственно, предпринимались попытки дать обоснование использования моделей типа (1) для описания популяционной динамики.

А. Н. Колмогоров [13] предполагал, что системы обыкновенных дифференциальных уравнений (ОДУ) применимы к описанию динамики популяций только тогда, когда численности последних достаточно велики. В этом случае имеется возможность ввести новые переменные для численности и времени, что соответствует «сжатию» исходных переменных, и при определенной степени сжатия позволит использовать системы ОДУ для аппроксимации значений новых переменных.

Если исходить из этого предположения, то тогда следует признать, что модели типа (1) непригодны для описания динамики разреженных популяций; да и что следует считать «достаточно большой численностью» популяции, а что нет – вопрос остается открытым. Как открытым остается и вопрос о том, существуют ли вообще такие популяции, к описанию динамики которых применимы ОДУ или нет. Понятно только одно: даже для массовых видов лесных насекомых в этом случае системы ОДУ практически неприменимы (если применимы, но не ко всему временному ряду) – в зоне стабильности и на фазе депрессии их численности крайне малы [2, 4].

А. А. Ляпуновым [15] был предложен иной подход к обоснованию правомерности использования детерминистских моделей к описанию динамики численности популяции. Он предполагал, что такие модели описывают не реальную динамику численности, что является случайным процессом, а динамику среднего числа особей в популяции. Действительно, если рассмотреть, к примеру, простой ветвящийся процесс Гальтона–Ватсона с непрерывным временем [16], то, как хорошо известно, состояние случайного процесса в каждый момент времени является случайной целочисленной величиной, в то время как среднее число особей удовлетворяет дифференциальному уравнению (1) с  $\beta = 0$ . Сходный результат мы имеем

также для процессов рождения и гибели. Таким образом, предполагалось, что детерминистские модели описывают динамику популяции в среднем, однако, строго это доказать (вывести соответствующие уравнения из вероятностных моделей) практически для всех моделей невозможно.

Если исходить из предположения А. А. Ляпунова, то тогда возникает весьма серьезная проблема, связанная с нахождением параметров модели. С одной стороны, нет никакой вероятностной модели, являющейся базовой для детерминистской, а, с другой, сравнение какой-либо одной случайной реализации со средним, по сути, лишено какого-либо смысла. Например, при сравнении случайной траектории надкритического простого ветвящегося процесса, которая «заходит» в нулевое состояние за конечное число шагов, с динамикой среднего (монотонно возрастающая экспонента) имеющиеся статистические критерии (критерий знаков, критерий Дарбина–Ватсона [17, 18]) покажут наличие зависимости в последовательности остатков. Это будет означать, что мы должны будем отклонить гипотезу о том, что модель пригодна для описания имеющихся экспериментальных данных.

Таким образом, в обоих случаях мы имеем проблемы, связанные с сопоставлением теоретических и экспериментальных данных. Или, точнее, в обоих случаях у нас нет оснований для использования экспериментальных данных для нахождения значений параметров модели. Для того, чтобы избежать проблемы сопоставления модельных и экспериментальных данных, мы должны согласиться с тем, что с помощью моделей типа (1) мы получаем некие (гладкие) кривые (решения задачи Коши для уравнения типа (1)), которые аппроксимируют экспериментальные данные по изменению численности популяции во времени. Именно аппроксимируют, но не являются численностью или плотностью популяции (хотя в некоторые моменты времени могут быть им равны).

Впрочем, можно называть переменную и численностью, и плотностью популяции. Кроме того, что переменная модели сопоставляется с реальной численностью, важно и то, что сама модель (для получения наилучшей аппроксимации) должна отражать соответствующим образом влияние различных регуляторов на численность (именно так, как эти регуляторы влияют на реальную численность популяции).



Однако признание переменной модели некоей «псевдочисленностью» еще не является достаточным условием для проведения сопоставления модельных и экспериментальных данных. Если априори предполагать, что динамика численности популяции является случайным процессом, то использование детерминистской модели для аппроксимации данных по одной-единственной реализации этого процесса (а для природных популяций мы принципиально не можем получить более одной реализации) выглядит весьма странно. Поэтому дополнительно мы обязаны предположить, что сама популяционная динамика *детерминирована* (подчиняется закону типа (1)), а отклонения экспериментальных данных от траекторий модели вызваны ошибками измерений.

Конечно, предположение о детерминированности популяционной динамики противоречит общепринятым представлениям о влиянии на флуктуации численности случайных факторов. Можно указать, например, на климатическую теорию [4, 19], в которой признаются наиболее важными регуляторами численности (насекомых) случайные погодные факторы. Однако, все разговоры о существенном влиянии на динамику популяций погодных факторов до настоящего времени не подкреплены какими-либо доказательствами (хотя бы на уровне статистической обработки информации). Имеется лишь некая совокупность наблюдений, которые собственно и служат основанием климатической теории.

**Какой статистический критерий правильный?** При оценке значений параметров математической модели необходимо использовать статистические критерии, которые позволяют не только оценить степень приближения имеющихся данных, но и сравнивать различные модели между собой (по степени приближения). Наиболее часто встречаемые критерии – это минимум суммы квадратов отклонений теоретических и экспериментальных данных в различных модификациях [17, 18, 20].

От выбора вида критерия, которому должны удовлетворять параметры модели, как и от выбора самой модели, существенно зависят результаты анализа. Выбор критерия, не соответствующего рассматриваемой проблеме, может не только исказить значения искомых параметров, но и привести в конечном итоге к неверным результатам. Однако, как определить какой именно критерий является правильным и соответствует рассматриваемым задачам, а какой нет?

С одной стороны, выбор статистического критерия зависит от постановки задачи, а, таким образом, зависит от главной цели проводимых исследований. Если мы хотим в результате исследований получить простой, надежный и достаточно точный прогноз изменения численности популяции за некий короткий интервал времени, то нам математическая модель в действительности не нужна (речь, естественно, идет о моделях, которые описывают популяционную динамику). В такой ситуации вполне можно обойтись известными стандартными методами статистики. И совсем другое дело, когда мы хотим оценить долговременные последствия воздействий на популяцию, или построить алгоритм определения оптимальной квоты вылова промысловой популяции – тут нам различные стандартные регрессионные методы фактически ничем помочь не могут. В этом случае мы должны знать реальные значения популяционных параметров.

С другой стороны, выбор статистического критерия нередко бывает лимитирован имеющимися возможностями. Например, отсутствие возможностей использовать современные вычислительные мощности приводят к тому, что критерии оценок параметров выбираются предельно простыми. В современной литературе нередко встречаются работы, в которых оценки параметров нелинейных систем дифференциальных уравнений, по сути, являются экспертными оценками: авторы не указывают ни доверительные интервалы для параметров, ни критерии оценки, ни вероятности реализации динамических режимов.

Пусть нам дана выборка  $\{\tilde{x}_k\}$ ,  $k=1, \dots, N$ , где  $\tilde{x}_k$  – численность рассматриваемой гипотетической популяции в  $t$ -ый момент времени,  $N+1$  – общее число наблюдений. Предположим для простоты, что выбранная нами модель имеет вид:

$$x_{k+1} = F(\bar{\alpha}, x_k), \quad (2)$$

где  $F(\cdot)$  – неотрицательная нелинейная функция, стоящая в правой части уравнения (2);  $\bar{\alpha}$  – вектор (неизвестных) параметров уравнения;  $x_k$  – численность (как было указано в предыдущем разделе;  $x_k$  является псевдо-численностью, но мы будем *по привычке* называть ее численностью) популяции в момент времени  $k$ ,  $k = 0, 1, 2, \dots$

Проблему можно поставить так: требуется оценить по имеющейся выборке  $\{\tilde{x}_k\}$  значения параметров рекуррентного уравнения (2). В этом случае можно воспользоваться, например, следующим критерием, который довольно часто встречается в литературе (см., например, [20–22]):

$$Q(\bar{\alpha}) = \sum_{t=0}^{N-1} (\tilde{x}_{t+1} - F(\bar{\alpha}, \tilde{x}_t))^2 \rightarrow \min_{\bar{\alpha}}. \quad (3)$$

Заметим, что, во-первых, если известны значения других переменных (если их в модели (2) несколько), то можно построить аналог выражения (3) и для такого более общего случая; во-вторых, наилучшие значения параметров получаются при достижении функционалом  $Q(\bar{\alpha})$  минимума, однако, учитывая нелинейность функции  $F$ , в общем случае таких минимумов может быть несколько.

Важно отметить, что при использовании критерия (3) уравнение (2) рассматривается не как динамическая модель, а исключительно как нелинейное регрессионное уравнение, предназначенное для прогнозирования изменения численности за шаг по времени. В этом случае возникает вполне естественный вопрос – а почему не взять другой критерий, аналогичный критерию (3), обеспечивающий наилучшую точность прогноза за два или большее число шагов по времени «вперед», используя при этом тоже уравнение (2)? Например, критерий, обеспечивающий наилучший прогноз на «два шага вперед», будет иметь вид

$$Q(\bar{\alpha}) = \sum_{t=0}^{N-2} (\tilde{x}_{t+2} - F^{(2)}(\bar{\alpha}, \tilde{x}_t))^2 \rightarrow \min_{\bar{\alpha}}, \quad (4)$$

где  $F^{(2)}(x) = F(F(x))$  – итерация функции  $F$ .

Заметим, что функционалы (3) и (4) существенно отличаются друг от друга как по числу слагаемых (у функционала (4) их меньше), так и по функции, стоящей в правой части. Поэтому в общем случае мы будем иметь различные оценки значений параметров модели (если, конечно, экспериментальные значения численности популяции не лежат строго на траектории модели (2)). Тогда почему значения, полученные с помощью критерия (3), мы должны считать истинными, а с помощью критерия (4) – нет?

Кроме этого, в выражении (3) (равно как и в выражении (4)) присутствуют «двойные стандарты»: к одному и тому же числу в данном выражении мы относимся по-разному. Действительно, в первом слагаемом в (3) величину  $\tilde{x}_1$  мы рассматриваем как значение некоей случайной величины, точнее, как некую константу, измеренную со случайной ошибкой. Однако уже во втором слагаемом мы считаем выражение  $F(\bar{\alpha}, \tilde{x}_1)$  теоретическим, значением численности популяции, которое теоретически должно было бы быть, если бы значение  $\tilde{x}_2$  было измерено без ошибки. Иными словами, во втором слагаемом мы считаем, что величина  $\tilde{x}_1$  измерена без ошибки.

С учетом сказанного, мы считаем, что критерии типа (3) к оценке параметров модели не применимы (для оценки истинных значений популяционных параметров, но применимы для построения прогнозов). Тем более неприменимы критерии, аналогичные по своей структуре критерию (3), но в которых используются измененные данные (прежде всего, после логарифмического преобразования (весьма популярного и широко используемого при статистической обработке биологических данных), и измененные соответствующим образом виды функций. Если относительно исходных данных у нас имеются некие основания считать, что ошибки измерений имеют нормальное распределение с нулевым средним, то после, например, логарифмического преобразования исходных данных, у нас такого основания нет.

Одним из возможных критериев выбора параметров модели является «глобальное приближение» (global fitting; [22]), при котором во всем множестве решений модели (2) ищется такое, которое лучше других приближает исходную эмпирическую траекторию. Обозначим через  $x_t^*(\bar{\alpha}, x_0)$  решение модели (2), которое получается при заданном векторе параметров  $\bar{\alpha}$  и заданным начальным значением переменной  $x_0$  (при таком подходе начальное значение также является неизвестным параметром, значение которого следует найти при помощи имеющихся экспериментальных данных). Тогда критерий примет следующий вид:

$$Q(\bar{\alpha}, x_0) = \sum_{k=0}^N (\tilde{x}_k - x_k^*(\bar{\alpha}, x_0))^2 \rightarrow \min_{\bar{\alpha}, x_0}. \quad (5)$$

Заметим, что в выражении (5) все значения исходной выборки в определенном смысле «равноправны» (отсутствуют «двойные стандарты», о которых речь шла выше). Можно также указать и другие «положительные стороны» критерия (5), отличающие его от критериев (3) и (4). В частности, если нам дана модель с несколькими переменными (паразиты, хищники, корм и так далее), а в эксперименте наблюдается только одна из них (например, только численность самой популяции  $\{\tilde{x}_k\}$ ), то с помощью критерия (5) можно оценить параметры всей модели, когда поведение траектории зависит от всех параметров модели. С другой стороны, этого принципиально нельзя сделать с помощью критериев (3) и (4), поскольку не все параметры модели входят в одно уравнение для численности популяции (в том числе, не входит интенсивность гибели паразитов, их плодовитость и так далее).

Кроме этого, если помимо выборки  $\{\tilde{x}_k\}$  известны значения других переменных, критерий (5) может быть легко модифицирован и для этого случая. Однако для огромного числа экспериментальных данных значений других переменных нет, либо существуют некие косвенные данные, имеющие непосредственное отношение к переменным модели. В то же время использование косвенных данных требует введения в модель неких дополнительных предположений, требующих отдельной проверки. Насколько же дополнительное предположение оправдано, как правило, не ясно, и в такой ситуации нельзя исключать возможности, когда постулирование дополнительной зависимости приводит к прямо противоположным конечным результатам.

Аналогичные суждения могут быть высказаны и в отношении другого косвенного показателя – уровня паразитизма, – который также может быть измерен в эксперименте (как и длина хвои) и который также требует дополнительных предположений для «пересчета» в численность паразитов. Известно, в частности, что этот показатель зависит не только от соотношения численностей популяций, но и от фазы градационного цикла [4], что в свою очередь потребует огромного числа дополнительных предположений для определения соответствующей численности паразитов. Эти предположения будут касаться, в первую очередь, идентификации для каждой отдельной точки фазы траектории массового размножения, что далеко не всегда может быть корректно проведено [2, 23, 24].

С учетом этого, при определении параметров модели рекомендуется использовать критерий (5). Некоторые авторы полагают [21, 25], что, если заключения, основанные на различных статистических критериях, совпадают, то это дает некую дополнительную уверенность в правильности полученных результатов. С этой точкой зрения трудно согласиться – совпадение или близость результатов, полученных с помощью одного «правильного» и группы «неправильных» критериев скорее должны настораживать, что где-то (в данных, алгоритмах и т. д.) имеется серьезная ошибка, чем служить обоснованием достоверности результатов.

**Заключение.** Таким образом, для обоснования применимости детерминистских моделей к описанию динамики численности биологических популяций необходимо постулировать, что численность популяции не является случайным процессом и ее изменение подчиняется детерминированному закону. Соответственно, следует считать все отклонения от этого закона только ошибками измерений. И, если это так, то тогда среднее значение отклонений должно равняться нулю, а само распределение отклонений должно быть в определенном смысле близко к нормальному.

Этого постулата, однако, недостаточно для правомерности использования детерминистских моделей равно как для аппроксимации экспериментальных данных по колебаниям численности популяций, так и для теоретических исследований популяционной динамики. Следует также признать, что существующие и используемые в настоящее время в экологическом моделировании модели типа (1) и (2) содержат переменную, которая не является ни численностью, ни плотностью популяции. Эта переменная является некоей псевдочисленностью, с помощью которой мы лишь пытаемся аппроксимировать динамику реальной численности.

Принятие постулата о детерминированном характере популяционной динамики, а также принятие соответствующего «взгляда» на основные переменные моделей позволяет не только обосновать применимость моделей к описанию динамики численности популяций и избежать проблем с обоснованием предположения о дифференцируемости переменной. Кроме этого, такие предположения, в частности, позволяют не дифференцировать экспериментальные данные на некие группы, для аппроксимации которых могут быть применимы детерминистские модели и для которых не могут быть применимы.

Сравнительный анализ различных статистических критериев, используемых в настоящее время для оценки значений параметров моделей, показал, что многие из них обладают негативными свойствами, по сути, исключающими их из совокупности критериев, с помощью которых можно оценить реальные значения популяционных параметров. Впрочем, это не исключает возможностей их использования при построении прогнозов изменения численностей. Указанных недостатков, – прежде всего, наличия «двойных стандартов», – нет только у «глобального приближения» (и некоторых его возможных модификаций), что позволяет рекомендовать именно этот критерий для оценки истинных значений популяционных показателей.

### Библиографический список

1. *Базыкин А. Д.* Математическая биофизика взаимодействующих популяций. М.: Наука, 1985. 181 с.
2. *Недорезов Л. В.* Моделирование массовых размножений лесных насекомых. Новосибирск: Наука, 1986. 125 с.
3. *Вольтерра В.* Математическая теория борьбы за существование. М.: Наука, 1976. 288 с.
4. Популяционная динамика лесных насекомых / А. С. Исаев, Р. Г. Хлебопрос, Л. В. Недорезов, В. В. Киселев, Ю. П. Кондаков, В. Г. Суховольский. М.: Наука, 2001. 347 с.
5. *Недорезов Л. В.* Курс лекций по математической экологии. Новосибирск: Сибирский хронограф, 1997. 161 с.
6. *Смит Дж. М.* Математические идеи в биологии. М.: Мир, 1970. 179 с.
7. *Смит Дж. М.* Модели в экологии. М.: Мир, 1976. 184 с.
8. *Свирижев Ю. М., Логофет Д. О.* Устойчивость биологических сообществ. М.: Наука, 1978. 352 с.
9. *Полужтков Р. А., Пых Ю. А., Швытов И. А.* Динамические модели экологических систем. Л.: Гидрометеиздат, 1980. 288 с.
10. *Уильямсон М.* Анализ биологических популяций. М.: Мир, 1975. 271 с.
11. *Volterra V.* Lecons sur la theorie mathematique de la lutte pour la vie. Paris: Gauthiers-Villars, 1931. 214 p.
12. *Lotka A. I.* Undamped oscillations derived from the law mass action // J. Amer. Chem. Soc., 1920, V. 42, № 8. P. 1595–1599.
13. *Kolmogoroff A. N.* Sulla theoria di Volterra della lotta per l'esistenza // Giornale dell'Inst Italiano degli attuary, 1936, V. 7. P. 74–80.

14. *Kostitzin V. A.* La Biologie Mathematique. Paris: A.Colin, 1937. 236 p.
15. *Ляпунов А. А., Багриновская Г. И.* О методологических вопросах математической биологии // Математическое моделирование в биологии, М.: Наука, 1975. С. 5–18.
16. *Севастьянов Б. А.* Ветвящиеся процессы. М.: Наука, 1971. 436 с.
17. *Дрейнер Н., Смит Г.* Прикладной регрессионный анализ. Том 1, М.: Финансы и статистика, 1986. 366 с.
18. *Дрейнер Н., Смит Г.* Прикладной регрессионный анализ. Том 2, М.: Финансы и статистика, 1987. 351 с.
19. *Uvarov B. P.* Insect and climate // Trans. Entomol. Soc., London, V. 79, 1931. P. 1–297.
20. *Лакин Г. Ф.* Биометрия. М.: Высш. шк., 1990. 352 с.
21. *Turchin P.* Complex Population Dynamics: A Theoretical/Empirical Synthesis. Princeton: Princeton University Press, 2003. 536 p.
22. *Wood S. N.* Partially specified ecological models // Ecological Monographs, V. 71, 2001. P. 1–25.
23. *Недорезов Л. В.* Фазовый портрет динамики численности *Panolis Flammea* Schiff // Известия СОАН СССР, сер. биол. наук, № 1, 1989. С. 81–86.
24. *Nedorezov L. V.* Restoration of Phase Portrait Structure for the Dynamics of a Forest Pest, the Pine Moth (*Dendrolimus pini* L.) // Ecol. Modell. 115, 1999. P. 35–44.
25. *Hilborn R., Mangel M.* The ecological detective. Princeton, New Jersey: Princeton University Press, 1997. 330 p.



**И. В. Панфилов**, доктор технических наук, профессор

## **ПРОБЛЕМНОЕ ОБУЧЕНИЕ**

Исходя из процесса формирования специалиста в высшей школе, следует, что лекция должна включать в себя, кроме конкретной информации, и элементы необходимые для включения механизмов восприятия, мышления и поведения.

Такому построению лекционного материала соответствует метод, используемый прием создания и решения проблемных ситуаций.

Проблемный метод обучения должен устранить недостатки традиционного метода обучения:

- пассивность обучения (слабое развитие психологии учения, приводящее к отождествлению учения-запоминания).
- рассчитан на относительно стабильную учебную информацию.

Таким образом, процесс усвоения учебной информации при существующей методике сводится к пониманию и запоминанию сообщаемой информации, воспроизведению и накоплению знаний. Сами обучаемые при таком методе фактически не принимают участия в активном добывании знаний.

Задача современной педагогической науки сводится к разработке методики, позволяющей научить студентов самостоятельно добывать и совершенствовать свои знания, вырабатывать у них способность применять свои знания, навыки и умения для решения практических вопросов, а также усилить процесс умственного развития обучаемых. Активизировать процесс обучения, что можно решить системой проблемного обучения.

«Проблемное обучение заключается в постоянном создании перед обучаемыми на занятиях проблемных ситуаций (проблемных задач) и разрешения их при максимальной самостоятельности и под направляющим руководством преподавателя. Процессы усвоения и применения знаний совершаются в ходе поисков решения проблемных ситуаций».

В каждой проблеме должно быть что-то неизвестно, в противном случае нет проблемы, нечего искать. Одновременно в проблеме должно быть что-то известно. Ибо, если не известно о требуемом предмете ничего, то и опознать его невозможно. В каждой проблеме должно быть условие, определяющее связи между известными признаками, элементами и неизвестными.

Преимуществом проблемного метода является и то, что, решая проблему, обучающийся не пренебрегает ранее приобретенными знаниями, а использует их. Он постоянно оперирует ими. У обучающегося усваивающего готовые знания (пассивный метод), они лежат «мертвым капиталом».

Следует иметь в виду, что все обучение осуществить только методом проблемных ситуаций невозможно, так как этот метод имеет не только преимущества, но и свои недостатки:

- он связан с большими затратами времени, особенно на начальном этапе;
- проблемное обучение в некоторых моментах является слабо управляемым, так как обучаемый при решении проблемы должен сам принимать решения. Отстающим студентам проблемы будут казаться неразрешимыми и они могут потерять интерес к обучению. Необходима дифференциация проблемных ситуаций для различных групп;
- при применении проблемного обучения возникают трудности в организации проблемно-группового обучения, то есть сочетания коллективного обучения с индивидуальной работой.

Таким образом, проблемное обучение нельзя рассматривать как метод решения всех проблем современного образования, как метод, который заменит традиционный метод обучения в высшей школе, и не может иметь универсального характера.

*Основные части проблемного обучения.* В проблемном обучении можно выделить 4 основные части:

- формулировка проблемы (осмысливание проблемной ситуации, понимание поставленной задачи);
- составление гипотез решения;
- решение проблемной ситуации (ПС);
- анализ полученных результатов.

*Формулировка проблемы* относится не столько к обучаемому, сколько к педагогу. Сформулировать проблемную ситуацию (ПС) в ряде случаев значительно труднее, нежели решить ее. После постановки проблемной ситуации обучаемый должен осмыслить, понять проблему. Не поняв проблемы, глупо пытаться решить ее. Если проблему не понимают, значит, она недостаточно четко сформулирована. Проблема должна быть естественной и интересной.

При осмысливании проблемной ситуации необходимо уяснить, что дано и что неизвестно, что нужно получить в результате решения проблемы.

Понимание проблемы составляет исходный пункт процесса самостоятельного мышления. Отсюда необходимость обязательного изучения педагогом уровня знаний обучаемого для обеспечения правильной формулировки проблемной ситуации.

*Составление гипотез решения проблемной ситуации.* Полагая, что проблема полностью осмыслена, необходимо оценить возможные пути ее решения, то есть составить ряд гипотез. Рождение гипотезы есть процесс рождения новых мыслей на основе знаний и опыта. Новые мысли есть результат либо действий, либо абстрактного мышления. Для составления гипотез обучаемый обязан включить в действие весь арсенал своих знаний, произвести анализ. Итак, для составления гипотез нужны знания. Хорошие идеи, гипотезы имеют своим источником прошлый опыт и ранее приобретенные знания.

Составление гипотезы начинается с оценки, не встречалась ли раньше подобная ПС и как ее решали. В чем отличие данной ПС от ранее встречавшейся?

**НЕПОДГОТОВЛЕННЫЙ СТУДЕНТ НЕ ЗАИНТЕРЕСУЕТСЯ НИКАКОЙ ПС, ибо любая ПС для него окажется чрезвычайно трудной.**

Если студенты не подготовлены к занятию, то никакое проблемное обучение не сделает его продуктивным, а время, отведенное на это занятие, **СЛЕДУЕТ СЧИТАТЬ БЕЗВОЗВРАТНО ПОТЕРЯННЫМ.**

*Решение ПС* является результатом анализа выбранных гипотез, отклонения ошибочных и выбора правильных. Решение проблемы самостоятельно позволяет получить прочные и глубокие знания, а также навыки использования этих знаний в новых ситуациях, в теории и практике. Если проблема не решается или она трудная, то педагог в процессе решения должен дать дополнительную информацию, необходимую для преодоления трудностей.

*Анализ полученных результатов.* Этап анализа полученных результатов позволяет формировать навыки целенаправленного анализа возможных решений, сравнения предыдущих решений с полученными.

Таким образом, четыре части проблемного обучения охватывают всю структуру акта учения.

**И. В. Терентьев**, доктор технических наук, профессор,  
**В. Ю. Неверовский**, зав. лабораторией древесных растений  
Ботанического сада ГЛТА,  
**М. А. Шубина**, кандидат технических наук,  
старший научный сотрудник

## **АЛГОРИТМЫ АВТОМАТИЧЕСКОГО ОПРЕДЕЛЕНИЯ ВОЗРАСТА РАСТЕНИЙ И ЖИВОТНЫХ ПО ФОТОГРАФИЯМ**

Определение возраста растений и животных по фотографиям – это метод неразрушающего контроля, особенно актуальный для редко встречающихся или исчезающих особей.

При выборе алгоритма автоматического определения возраста живых организмов мы руководствовались следующими естественными ограничениями:

- алгоритм не должен быть эвристическим – то есть он не должен содержать операции или математические модели, для которых невозможно проверить их адекватность реальным процессам или явлениям;
- алгоритм должен иметь механизм самообучения на обучающем множестве произвольного размера.

Алгоритмы, удовлетворяющие сформулированным двум условиям, – это адаптивные и неадаптивные нейронные сети.

Обучающее множество для растений может предоставить дендрологический сад Санкт-Петербургской государственной лесотехнической академии, являющийся одним из старейших дендрологических садов России, в 2008 г. ему исполняется 175 лет.

Дендрологический сад, является составной частью Ботанического сада ГЛТА, заложен в 1833 г. в честь 30-летия основания Лесного института, сейчас Санкт-Петербургской государственной лесотехнической академии им. С. М. Кирова [1–3].

Создавалась коллекция известными дендрологами: Р. И. Шредером, Э. Л. Вольфом, В. Н. Сукачевым, П. А. Акимовым, Н. М. Андроновым, П. Л. Богдановым, Н. О. Соколовым, Н. Е. Булыгиным.

В настоящее время площадь дендросада составляет около 8 гектар.

Коллекция содержит более 1000 таксонов растений интродуцентов, является одной из самых богатых коллекций в северных широтах. В коллекции широко представлены растения дендрофлоры России, а также Северной Америки, Китая, Японии и т.д.

Коллекция древесных растений служит учебной базой лесохозяйственного факультета, объектом научных исследований, представляет собой культурно-историческую ценность.

В качестве обучающего множества выбраны представители семейства *Pinaceae* (Сосновые): пихта, ель, лжетсуга, их виды и разновидности, представленные различными возрастными группами.

Рассмотрим несколько алгоритмов автоматического определения возраста растений и животных по фотографиям.

### 1. Неадаптивный искусственный нейрон

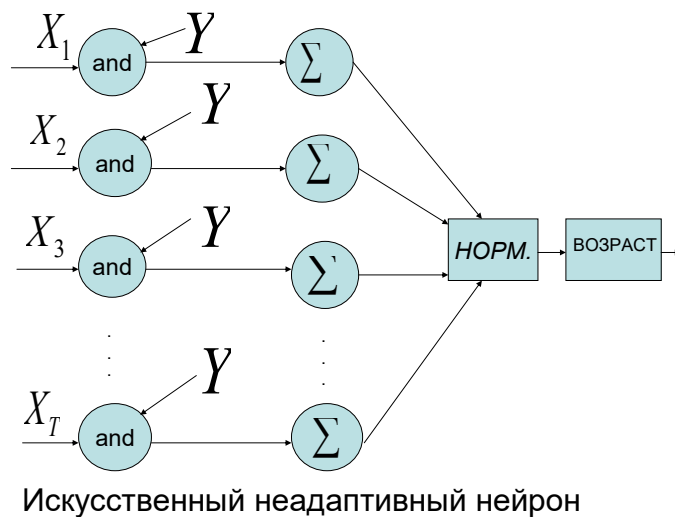
На каждое анализируемое изображение объекта последовательно «накладываются» все эталонные изображения с известным возрастом особи (составляющие обучающее множество), и для каждого эталона подсчитывается число голосов (совпадений яркостей  $k$ -го анализируемого и  $i$ -го эталонного изображения) в его пользу. После подсчета голосов для всех  $R$  эталонов эти голоса нормируются к единице – то есть определяются вероятности (геометрические)  $P_i^k$  ( $i = 1, 2, \dots, R$ ) для всех  $R$  эталонов. Решение – возраст особи, соответствующий  $k$ -му изображению определяется в виде

$$OLD1_k \cong \sum_{i=1}^{i=R} P_i^k \cdot OLD_i.$$

Здесь  $OLD_i$  – возраст особи, соответствующий  $i$ -му эталонному изображению;

$OLD1_k$  – возраст особи, соответствующий  $k$ -му анализируемому изображению.

Алгоритм формирования решения можно представить схемой искусственного неадаптивного нейрона рис. 1.



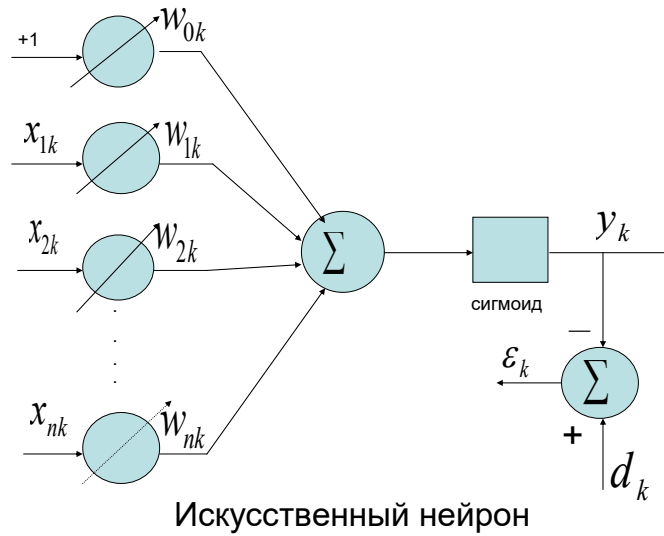
**Рис. 1.** Структура искусственного неадаптивного нейрона

$T$  – количество анализируемых изображений;  $X_k$  – множество из  $n$  пикселей  $k$ -го анализируемого изображения;  $Y = (Y_1, Y_2, \dots, Y_R)$ , где  $Y_i$  – множество из  $n$  пикселей, соответствующих изображению  $i$ -го эталона ( $i = 1, 2, \dots, R$ );  $\Sigma$  – суммирование количества совпадений яркостей пикселей в множествах  $X_k$  и  $Y_i$ ; *НОРМ.* – нормировка к единице голосов для всех объектов-эталонов, то есть вычисление вероятностей (геометрических)  $P_i^k$  ( $i = 1, 2, \dots, R$ ). *ВОЗРАСТ* – определение возраста особи, соответствующей  $k$ -му изображению  $OLD1_k \cong \sum_{i=1}^{i=R} P_i^k \cdot OLD_i$ .

## 2. Адаптивный искусственный нейрон [4–6]

Структура искусственного адаптивного нейрона представлена на рис. 2.

На рисунке обозначено:  $X_k = (x_{1k}, x_{2k}, \dots, x_{nk})$ ; ( $k = 1, 2, \dots, T$ ) – обучающие (эталонные) объекты (изображения из  $n$  пикселей с яркостями  $x_{1k}, x_{2k}, \dots, x_{nk}$ );  $W_k = (w_{1k}, w_{2k}, \dots, w_{nk})$  – весовые коэффициенты на  $k$ -м этапе обучения;  $y_k$  – выход нейрона на  $k$ -м этапе обучения;  $d_k$  – желаемый выход нейрона на  $k$ -м этапе обучения;  $\varepsilon_k = d_k - y_k$  – ошибка на выходе нейрона на  $k$ -м этапе обучения, служащая для подстройки весовых коэффициентов на  $k$ -м этапе обучения.



**Рис. 2.** Структура адаптивного искусственного нейрона

Сигмоид (рис. 2) осуществляет преобразование входного сигнала  $s$  следующим образом:

$$y = \frac{1}{1 + e^{-s}} \quad (1)$$

Выразив  $e^{-s}$  из (1):  $e^{-s} = \frac{1-y}{y}$ , получим выражение для производной:

$$\frac{dy}{ds} = \frac{e^{-s}}{[1 + e^{-s}]^2} = y(1-y) \quad (2)$$

Для изменения (подстройки) вектора весов  $W_k = (w_{0k}, w_{1k}, \dots, w_{nk})$  после  $k$ -го вычисления выходного значения  $y_k$  нейрона используется метод наискорейшего спуска (методика Backpropagation – обратного распространения ошибки):

$$W_{k+1} = W_k + \mu(-\nabla_k). \quad (3)$$

Здесь  $0 < \mu \leq 1$  – коэффициент, контролирующий стабильность и скорость сходимости;  $\nabla_k = \frac{\partial \varepsilon_k^2}{\partial W_k} = 2\varepsilon_k \frac{\partial \varepsilon_k}{\partial W_k}$  – градиент ошибки  $\varepsilon_k = d_k - y_k$ .

Для линейного выхода нейрона (до сигмоидальной функции) в соответствии с рис. 2 имеем:

$$s_k = w_{0k} + \sum_{i=1}^n x_{ik} w_{ik}. \quad (4)$$

Выход нейрона в соответствии с (1) представляется так:

$$y_k = \frac{1}{1 + e^{-s_k}}. \quad (5)$$

Так как по (2)  $\frac{\partial \varepsilon_k}{\partial w_{ik}} = \frac{\partial \varepsilon_k}{\partial y_k} \frac{\partial y_k}{\partial s_k} \frac{\partial s_k}{\partial w_{ik}} = -y_k(1 - y_k)x_{ik}$ , то для компонент градиента получим  $\nabla_{ik} = 2\varepsilon_k \frac{\partial \varepsilon_k}{\partial w_{ik}} = -2\varepsilon_k y_k(1 - y_k)x_{ik}$ .

Поэтому в соответствии с (3) находим поправки для компонент вектора весов после  $k$ -го вычисления  $y_k$ :

$$w_{i,k+1} = w_{ik} + 2\mu(d_k - y_k)y_k(1 - y_k)x_{ik} \quad (i = 0, 1, \dots, n). \quad (6)$$

Начальное значение для нулевой компоненты вектора весов равно

$$w_{01} = 1.$$

И в соотношении (6) предполагается также равенство  $x_{0k} = 1$ .

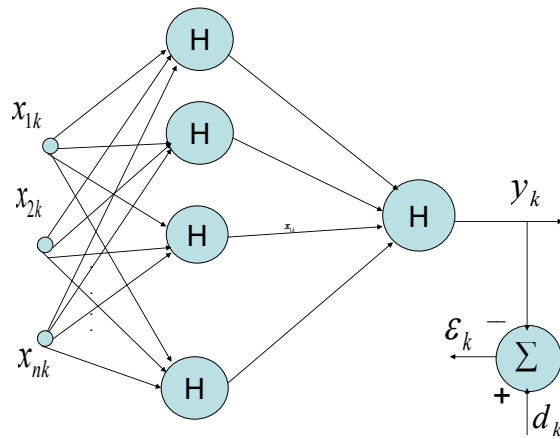
В качестве начальных значений для остальных компонент вектора весов  $w_{i1}$  ( $i = 1, 2, \dots, n$ ) выбираются небольшие случайные числа из диапазона  $(-0,1 - 0,1)$ . Значение  $y_k$  в (6) определяется соотношениями (4), (5).

### 3. Адаптивная двухслойная нейронная сеть

На рис.3 представлена двухслойная нейронная сеть.

Первый слой состоит из  $\nu$  нейронов. Второй слой состоит из одного нейрона. На входы каждого входного нейрона подается обучающий вектор  $X_k = (x_{1k}, x_{2k}, \dots, x_{nk})$  –  $k$ -е эталонное изображение из  $n$  пикселей с яркостями  $x_{1k}, x_{2k}, \dots, x_{nk}$ .





**Рис. 3.** Двухслойная нейронная сеть

Второй слой состоит из одного нейрона на входы которого подаются  $v$  выходов нейронов первого слоя. На линейном выходе  $r$ -го нейрона первого слоя имеем:

$$s_{rk}^{(1)} = w_{0rk}^{(1)} + \sum_{i=1}^n x_{ik} w_{irk}^{(1)}, \quad (r = 1, 2, \dots, v). \quad (7)$$

А на нелинейном выходе  $r$ -го нейрона первого слоя получим:

$$y_{rk}^{(1)} = \frac{1}{1 + e^{-s_{rk}^{(1)}}} \quad (r = 1, 2, \dots, v). \quad (8)$$

Нелинейные выходы (8) нейронов первого слоя являются входами нейрона второго слоя. На линейном выходе нейрона второго слоя образуется сумма

$$s_k^{(2)} = w_{0k}^{(2)} + \sum_{i=1}^v y_{ik}^{(1)} w_{ik}^{(2)}. \quad (9)$$

Нелинейный выход нейрона второго слоя

$$y_k^{(2)} = \frac{1}{1 + e^{-s_k^{(2)}}} = y_k \quad (10)$$

является выходом нейронной сети.

В соответствии с методикой Backpropagation, учитывая, что

$$\varepsilon_k = (d_k - y_k), \quad (11)$$

определяем градиенты для весовых коэффициентов второго слоя

$$\nabla_{ik}^{(2)} = 2\varepsilon_k \frac{\partial \varepsilon_k}{\partial w_{ik}^{(2)}} = 2\varepsilon_k \frac{\partial \varepsilon_k}{\partial y_k} \frac{\partial y_k}{\partial s_k^{(2)}} \frac{\partial s_k^{(2)}}{\partial w_{ik}^{(2)}} = -2\varepsilon_k y_k (1 - y_k) y_{ik}^{(1)}.$$

Поэтому поправки для коэффициентов второго слоя имеют вид:

$$w_{i,k+1}^{(2)} = w_{i,k}^{(2)} + 2\mu \varepsilon_k y_k (1 - y_k) y_{ik}^{(1)}, \quad (i = 0, 1, 2, \dots, \nu). \quad (12)$$

Начальное значение для нулевой компоненты вектора весов равно:

$$w_{01}^{(2)} = 1.$$

В качестве начальных значений для остальных компонент вектора весов  $w_{01}^{(2)} = 1$  ( $I = 1, 2, \dots, \nu$ ) выбираются небольшие случайные числа из диапазона  $(-0,1 - 0,1)$ . В соотношении (12) предполагается также равенство  $y_{0k}^{(1)} = 1$ .

Значение  $y_{ik}^{(1)}$  ( $I = 1, 2, \dots, \nu$ ) в (12) определяется соотношениями (7), (8).

В соответствии с методикой Backpropagation определяем градиенты для весовых коэффициентов нейронов первого слоя.

$$\nabla_{irk}^{(1)} = 2\varepsilon_k \frac{\partial \varepsilon_k}{\partial w_{irk}^{(1)}} = 2\varepsilon_k \frac{\partial \varepsilon_k}{\partial y_k} \frac{\partial y_k}{\partial s_k^{(2)}} \frac{\partial s_k^{(2)}}{\partial y_{rk}^{(1)}} \frac{\partial y_{rk}^{(1)}}{\partial s_{rk}^{(1)}} \frac{\partial s_{rk}^{(1)}}{\partial w_{irk}^{(1)}}, \quad (i = 1, 2, \dots, n); \quad (r = 1, 2, \dots, \nu) \quad (13)$$

В соотношении (13)

$$\frac{\partial \varepsilon_k}{\partial y_k} = -1; \quad \frac{\partial y_k}{\partial s_k^{(2)}} = y_k (1 - y_k); \quad \frac{\partial s_k^{(2)}}{\partial y_{rk}^{(1)}} = w_{rk}^{(2)}; \quad \frac{\partial y_{rk}^{(1)}}{\partial s_{rk}^{(1)}} = y_{rk}^{(1)} (1 - y_{rk}^{(1)}); \quad \frac{\partial s_{rk}^{(1)}}{\partial w_{irk}^{(1)}} = x_{ik}. \quad (14)$$

Поэтому для коэффициентов первого слоя получим:

$$w_{ir,k+1}^{(1)} = w_{ir,k}^{(1)} + \mu(-\nabla_{irk}^{(1)}) = w_{ir,k}^{(1)} + 2\mu\varepsilon_k y_k (1 - y_k) w_{rk}^{(2)} y_{rk}^{(1)} (1 - y_{rk}^{(1)}) x_{ik}, \quad (15)$$

$(r=1,2,\dots,v); (i=0,1,2,\dots,n)$ .

Начальные значения для нулевых компонентов вектора весов равны

$$w_{0r,1}^{(1)} = 1; (r=1,2,\dots,v)$$

В качестве начальных значений для остальных компонент вектора весов  $w_{ir,1}^{(1)} = 1$   $(i=1,2,\dots,n); (r=1,2,\dots,v)$  выбираются небольшие случайные числа из диапазона  $(-0,1 - 0,1)$ . В соотношении (15) предполагается также выполнение равенства  $x_{0k}^{(1)} = 1$ .

Значение  $\varepsilon_k$  в (15) определяется соотношением (11), а  $y_k, w_{rk}^{(2)}$  – по (9), (12).

Значение  $y_{rk}^{(1)}$   $(r=1,2,\dots,v)$  в (15) определяется соотношениями (7), (8).

#### 4. Адаптивная трехслойная нейронная сеть

На рис. 4 изображена схема соединений трехслойной нейронной сети.

В первом и втором слое содержится по  $v$  нейронов. Третий слой состоит из одного выходного нейрона.

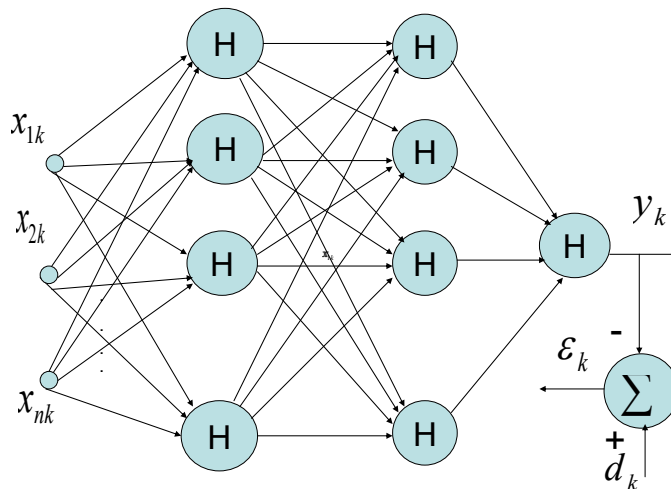


Рис. 4. Трехслойная нейронная сеть

Первый и второй слой нейронов состоит из  $v$  нейронов. На входы каждого нейрона первого слоя подается обучающий вектор  $X_k = (x_{1k}, x_{2k}, \dots, x_{nk})$  –  $k$ -е эталонное изображение из  $n$  пикселей с яркостями  $x_{1k}, x_{2k}, \dots, x_{nk}$ . Третий слой состоит из одного нейрона на входы которого подаются  $v$  выходов нейронов второго слоя. Для линейного выхода  $r$ -го нейрона первого слоя справедливо соотношение (7), а для нелинейного выхода  $r$ -го нейрона первого слоя справедливо соотношение (8).

Нелинейный выход  $r$ -го нейрона первого слоя (8) является  $r$ -м компонентом входного вектора для нейронов второго слоя. На линейном выходе нейрона  $t$  второго слоя образуются суммы

$$s_{ik}^{(2)} = w_{0ik}^{(2)} + \sum_{i=1}^v y_{ik}^{(1)} w_{ik}^{(2)}, \quad (t = 1, 2, \dots, v). \quad (16)$$

Нелинейный выход нейрона  $t$  второго слоя

$$y_{ik}^{(2)} = \frac{1}{1 + e^{-s_{ik}^{(2)}}}, \quad (t = 1, 2, \dots, v) \quad (17)$$

является входом нейрона третьего слоя.

На линейном выходе нейрона третьего слоя образуется сумма

$$s_k^{(3)} = w_{0k}^{(3)} + \sum_{i=1}^v y_{ik}^{(2)} w_{ik}^{(3)}. \quad (18)$$

При этом нелинейный выход нейрона третьего слоя определяется следующим образом:

$$y_k^{(3)} = \frac{1}{1 + e^{-s_k^{(3)}}} = y_k. \quad (19)$$

Повторяя рассуждения при выводе соотношения (12), получим поправки для коэффициентов нейрона третьего слоя

$$w_{i,k+1}^{(3)} = w_{i,k}^{(3)} + 2\mu \varepsilon_k y_k (1 - y_k) y_{ik}^{(2)}, \quad (i = 0, 1, 2, \dots, v). \quad (20)$$

Начальное значение для нулевой компоненты вектора весов равно:

$$w_{01}^{(3)} = 1;$$

В качестве начальных значений для остальных компонент вектора весов  $w_{01}^{(3)} = 1$ ; ( $i = 1, 2, \dots, v$ ) выбираются небольшие случайные числа из диапазона  $(-0,1 - 0,1)$ . В соотношении (20) предполагается также равенство  $y_{0k}^{(2)} = 1$ .

Для градиентов второго слоя имеем:

$$\nabla_{itk}^{(2)} = 2\varepsilon_k \frac{\partial \varepsilon_k}{\partial w_{itk}^{(2)}} = 2\varepsilon_k \frac{\partial \varepsilon_k}{\partial y_k} \frac{\partial y_k}{\partial s_k^{(3)}} \frac{\partial s_k^{(3)}}{\partial y_{tk}^{(2)}} \frac{\partial y_{tk}^{(2)}}{\partial s_{tk}^{(2)}} \frac{\partial s_{tk}^{(2)}}{\partial w_{itk}^{(2)}} = -2\varepsilon_k y_k (1 - y_k) w_{tk}^{(3)} y_{tk}^{(2)} (1 - y_{tk}^{(2)}) y_{ik}^{(1)}$$

$(i = 1, 2, \dots, v); (t = 1, 2, \dots, v).$  (21)

Поэтому поправки для коэффициентов второго слоя определяются следующим образом:

$$w_{it,k+1}^{(2)} = w_{it,k}^{(2)} + 2\mu \varepsilon_k y_k (1 - y_k) w_{tk}^{(3)} y_{tk}^{(2)} (1 - y_{tk}^{(2)}) y_{ik}^{(1)},$$

$(i = 0, 1, 2, \dots, v); (t = 1, 2, \dots, v).$  (22)

Начальные условия для коэффициентов нейрона  $t$  второго слоя:

$$w_{0t1}^{(2)} = 1, (t = 1, 2, \dots, v). \quad (23)$$

И предполагается, что  $y_{0k}^{(1)} = 1$ .

В качестве начальных условий для остальных компонент коэффициентов нейрона  $t$  второго слоя  $w_{it1}^{(2)}$  ( $i = 1, 2, \dots, v$ ); ( $t = 1, 2, \dots, v$ ) выбираются небольшие случайные числа из диапазона  $(-0,1 - 0,1)$ .

Для производной  $\frac{\partial \varepsilon_k}{\partial w_{irk}^{(1)}}$  получаем:

$$\frac{\partial \varepsilon_k}{\partial w_{irk}^{(1)}} = \frac{\partial \varepsilon_k}{\partial y_k} \frac{\partial y_k}{\partial s_k^{(3)}} \sum_{l=1}^v \frac{\partial s_k^{(3)}}{\partial y_{lk}^{(2)}} \frac{\partial y_{lk}^{(2)}}{\partial s_{lk}^{(2)}} \frac{\partial s_{lk}^{(2)}}{\partial y_{rk}^{(1)}} \frac{\partial y_{rk}^{(1)}}{\partial s_{rk}^{(1)}} \frac{\partial s_{rk}^{(1)}}{\partial w_{irk}^{(1)}} =$$

$$= -y_k(1-y_k)y_{rk}^{(1)}(1-y_{rk}^{(1)})x_{ik} \sum_{l=1}^v w_{lk}^{(3)} y_{lk}^{(2)} (1-y_{lk}^{(2)}) w_{rk}^{(2)}$$

Поэтому поправки для коэффициентов  $w_{irk}^{(1)}$  принимают вид:

$$w_{ir,k+1}^{(1)} = w_{irk}^{(1)} + 2\mu(d_k - y_k)y_k(1-y_k)y_{rk}^{(1)}(1-y_{rk}^{(1)})x_{ik} \sum_{l=1}^v w_{lk}^{(3)} y_{lk}^{(2)} (1-y_{lk}^{(2)}) w_{rk}^{(2)}$$

$$(i=0, 1, 2, \dots, n); (r=1, 2, \dots, v). \quad (24)$$

И здесь предполагается, что  $x_{0k} = 1$ .

Начальные условия для  $w_{irk}^{(1)}$ :

$$w_{0r1}^{(1)} = 1; (r=1, 2, \dots, v).$$

В качестве начальных значений для остальных компонент вектора весов  $w_{ir1}^{(1)}, \dots$  ( $i=1, 2, \dots, n$ ); ( $r=1, 2, \dots, v$ ) выбираются небольшие случайные числа из диапазона  $(-0,1 - 0,1)$ .

## 5. Особенности режима обучения адаптивной нейронной сети

1. На  $n$  входов каждого из входных нейронов подаются  $T$  раз обучающие выборки  $X_k = (x_{1k}, x_{2k}, \dots, x_{nk})$  – эталонные изображения из  $n$  пикселей с яркостями  $x_{1k}, x_{2k}, \dots, x_{nk}$ ; ( $k=1, 2, \dots, T$ ). Обучение одного нейрона проводится в соответствии с (6).

Обучение двухслойной сети соответствует соотношениям (12), (15).

Обучение трехслойной сети проводится в соответствии с (20), (22), (24).

2. Поправки для весовых коэффициентов (6), (12), (15), (20), (22), (24) для каждого из эталонных изображений определяются один раз.

3. Желаемым выходом  $d_k$  для всех эталонных изображений является величина  $d_k = 1$ .

## 6. Особенности вычислительных процессов при использовании адаптивных нейронных сетей

На вход нейронной сети по окончании периода обучения (то есть после определения поправок для весовых коэффициентов) подается анализируемое изображение особи. При этом последовательно используются весовые коэффициенты для каждого из эталонных объектов. Выходом нейронных сетей для  $k$ -го анализируемого изображения являются вероятности  $y_k = P_i^k (i = 1, 2, \dots, R)$ .

Возраст особи, соответствующей  $k$ -му изображению определяется аналогично неадаптивному нейрону:  $OLD1_k \cong \sum_{i=1}^{i=R} P_i^k \cdot OLD_i$ .

## 7. Некоторые результаты численных экспериментов

Приведенные выше алгоритмы и соответствующие им программы испытывались на 28 представителях хвойных пород Ботанического сада ГЛТА (диапазон возрастов 12–108 лет). Для съемки использовалась цифровая камера *FinePix S9600*. Испытание проводилось следующим образом: последовательно исключалось из эталонных первое, второе, ..., 28 изображение. Оставшиеся 27 изображений считались эталонными, а для исключенного изображения определялся возраст соответствующего растения. Подсчитывалась абсолютная ошибка для каждого определения возраста и средняя абсолютная ошибка. Средняя абсолютная ошибка для неадаптивного искусственного нейрона составила 13,8 года, а для трехслойной нейронной сети с 30 нейронами в первом и втором слое 13,3 года.

Приведенные выше алгоритмы и соответствующие им программы могут быть использованы для определения возраста многих живых организмов. В частности, мы провели испытания алгоритмов при решении задачи определения возраста сотрудников кафедры информатики и информационных систем, некоторых сотрудников других кафедр и студентов по изображениям, полученным в результате съемки на пленочную камеру-автомат Cannon EOS 3000v. Для ввода изображений в PC использовался сканер Epson 4990 при использовании глубины цвета 48 бит на пиксел, что дает при использовании модели RGB 16 бит на пиксел для каждой цвето-

вой компоненты RED, GREEN, BLUE. В качестве эталонных использовались 31 изображения и 4 канала: GREY, RED, GREEN, BLUE.

Полученная таким образом средняя абсолютная ошибка для неадаптивного искусственного нейрона составила 14,5 года, а для трехслойной нейронной сети с 30 нейронами в первом и втором слое 14,8 года. Эта ошибка может быть снижена за счет уменьшения диапазона исследуемых возрастов (в эксперименте диапазон возрастов сотрудников составил 22–83 года) и / или увеличения количества эталонов в обучающей выборке.

### Библиографический список

1. *Безбах С. А.* Лесной. Ч. 1. Л., 1929. (Отделение Лен. окр. общ-ва краеведения на финско-ладожском перешейке. Сер. «Выборгский район». Вып. 1).
2. Исторический очерк развития Санкт-Петербургского лесного института (1803–11903) / Сост. П. Н. Вереха, М. М. Орлов. Сп-б., 1903.
3. Инвентаризационное описание Ботанического сада Ленинградской ордена Ленина Лесотехнической академии им. С. М. Кирова. / С-3 Лесостроительное предприятие. Всесоюзное объединение «Леспроект», 1984.
4. *Widrow B., Lehr M. A.* 30 Years of Adaptive Neural Networks: Perceptron, Madaline, and Backpropagation // Proceedings of the IEEE, vol. 78, No. 9, September 1990, p. 1415–1442.
5. *Тарков М. С.* Нейрокомпьютерные системы: Учебное пособие. М.: Интернет-Университет информационных технологий; БИНОМ. Лаборатория знаний, 2006. 142 с.
6. *Осовский С.* Нейронные сети для обработки информации. М.: Финансы и статистика. 2004–344 с.



**И. В. Терентьев**, доктор технических наук, профессор,  
**М. А. Шубина**, кандидат технических наук, старший научный сотрудник,  
**Г. А. Морус**, инженер лесного хозяйства, аудитор ООО «Европартнер»

## **УСОВЕРШЕНСТВОВАННЫЙ АЛГОРИТМ КЛАССИФИКАЦИИ ЛЕСОХОЗЯЙСТВЕННЫХ ОБЪЕКТОВ НА АЭРОКОСМИЧЕСКИХ ИЗОБРАЖЕНИЯХ ЗЕМНОЙ ПОВЕРХНОСТИ**

Автоматизированная обработка аэрокосмических (АК) изображений может применяться для хозяйственного и экологического мониторинга лесов, стратификации территории лесного фонда при инвентаризации малоосвоенных и неосвоенных лесов и их мелкомасштабном картографировании. Использование в этих целях традиционных визуальных методов анализа и интерпретации АК-изображений имеет необходимое методическое обеспечение и развитую практику применения. Однако визуальные методы не позволяют проводить мониторинг, стратификацию и картографирование лесного фонда на обширных территориях с необходимой периодичностью, прежде всего из-за высокой трудоемкости. Особое значение приобретает разработка автоматизированных методов обработки АК-изображений в связи с интенсивным развитием новых информационных технологий анализа и представления пространственно распределенных данных – геоинформационных технологий.

Временные (стоимостные) затраты при использовании автоматизированных систем обработки тем ниже, а их производительность тем выше, чем большее число операций выполняются в автоматическом режиме. Поэтому весьма желательна автоматизация хотя бы части не полностью автоматизированных операций.

В работе [1] рассмотрен алгоритм надежного различения объектов – эталонов, гарантирующий их достоверное различение при произвольном количестве и взаимном перекрытии сигнатур, в котором для классификации используется вариант метода многоальтернативного дискретного накопления [2]. В работе [3] этот метод обобщен на задачу построения иерархической системы понятий.

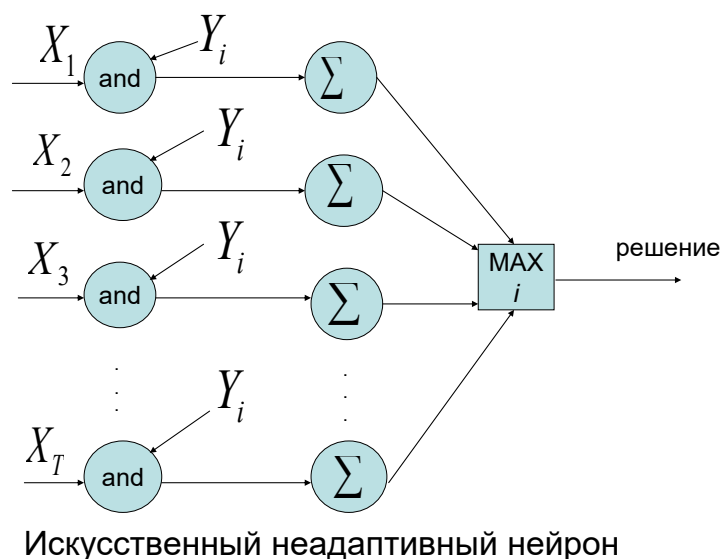
В работе [4] исследовалась возможность автоматического выделения (картографирования) лесохозяйственных объектов с помощью алгоритма [1]. Безошибочное различение объектов – эталонов по методу [1] в общем случае не гарантирует достоверное выделение (картографирование) объектов, не входящих в обучающую выборку. Поэтому в [4] определялись условия, при которых алгоритм [1] безошибочного различения объектов – эталонов обеспечивал и безошибочное выделение (картографирование) объектов, не входящих в обучающую выборку, и построен соответствующий алгоритм

Дальнейшие исследования [5] показали, что вариант метода многоальтернативного дискретного накопления [1] может быть усовершенствован, и такое усовершенствование может быть использовано для классификации двух градационных объектов, различающихся по форме.

Дальнейшие усовершенствования алгоритма [5], привели к решению задачи классификации полутонных цветных изображений с помощью двухэтапного голосования.

При испытании алгоритма использовались изображения с глубиной цвета 48 бит на пиксель, что обеспечило для каждого канала RED, GREEN, BLUE 65536 градаций яркости; использовались также полусуммы изображений RED + GREEN, RED + BLUE, GREEN + BLUE, а также «серый» канал GREY – всего 7 каналов. Различение объектов по этому методу сводится к следующим двум этапам.

**Голосование первого этапа.** Все 7 каналов (изображений) рассматриваются как один вектор. Для каждого пикселя изображения определяется множество пикселей, попадающих в зрачок анализирующей системы с центром в этом пикселе, яркости и координаты яркостей этого множества. Затем последовательно «накладываются» все эталоны на отобранное множество и для каждого эталона подсчитывается число голосов в его пользу. После подсчета всех голосов выбирается решение в пользу того класса, для которого количество голосов максимально. Соответствующее решение выводится – анализируемый пиксель подсвечивается условным цветом. Если при этом получается нулевая сумма, то этот вариант считается отказом от классификации и подсвечивается яркостью белого цвета. Алгоритм формирования изображения решения после первого голосования можно представить схемой искусственного неадаптивного нейрона рис. 1.



**Рис. 1.** Структура искусственного неадаптивного нейрона

Структура соответствует голосованию первого этапа.  $T$  – количество пикселей изображения;  $X_k$  – множество пикселей, попадающих в зрачок системы с центром в  $k$ -й точке изображения;  $Y_i$  – множество пикселей, соответствующих  $i$ -му эталону;  $\Sigma$  – суммирование количества совпадений яркостей пикселей в множествах  $X_k$  и  $Y_i$ ;  $MAX_i$  – выбор в качестве решения для  $k$ -й точки изображения того эталона и соответствующего ему класса для которого количество совпадений яркостей пикселей в множествах  $X_k$  и  $Y_i$  максимально.

**Голосование второго этапа.** На изображении, полученном в результате голосования первого этапа, в скользящем квадрате заданного размера подсчитывается число голосов в пользу каждого условного цвета. Выбирается тот цвет, для которого число голосов максимально. Этим цветом и подсвечивается центр скользящего квадрата.

Были проведены экспериментальные расчеты для аэрофотоснимка масштаба 1:25000 территории Майского лесничества Гусевского лесхоза.

Схема категорий земель для участка этой территории представлена на рис. 2.



**Рис. 2.** Схема категорий земель части территории Майского лесничества Гусевского лесхоза Калининградской области

Здесь можно выделить два крупных класса: сенокосы (суходольные – контур 14 и заболоченные – контур 26) и лес (контур 1–13, 15–25). Для площадей, покрытых лесом, преобладающими являются разные породные группы: хвойные, твердолиственные и мягколиственные. К хвойным группам относятся контуры 6, 8, 10, 11, 21. К твердолиственным группам (дуб) относятся контуры 4, 7. К мягколиственным группам (береза, липа, осина, ольха черная) принадлежат контуры 1–3, 5, 9, 12, 13, 15–20, 22–25.

Дальнейшее разделение этих групп можно связать с их таксационными характеристиками по соответствующим контурам рис. 2, и их возможные значения (применительно к лесному фонду Калининградской области).

#### ***Список категорий земель***

1. Покрытые древесной растительностью (леса).
2. Покрытые кустарниковой растительностью (кустарники).
3. Несомкнувшиеся лесные культуры.
4. Вырубки (необлесившиеся).
5. Прогалины, пустыри.
6. Погибшие насаждения, гари, ветровальники.
7. Сады фруктовые (искусственного происхождения).
8. Сенокосы, пастбища, выгоны.
9. Пашни, огороды.

10. Болота.
11. Пески.
12. Дороги, карьеры.
13. Трассы, просеки, канавы.
14. Поселки, усадьбы.
15. Воды (реки, озера, пруды).

***Показатели для категории земель, покрытых древесной растительностью (лесами)***

1. Преобладающие группы древесных пород: хвойные (сосна, ель), твердолиственные (дуб, ясень, граб), мягколиственные (береза, липа, ольха черная, осина).

2. Преобладающие древесные породы: сосна, ель, дуб, ясень, граб, береза, липа, ольха черная, осина.

3. Группы состава насаждений по группам пород:

- подавляющее преобладание какой-либо одной породной группы (8 единиц состава и более),
- простое преобладание какой-либо одной породной группы (5–7 единиц состава),
- примерно равное участие породных групп в составе насаждений.

4. Группы состава насаждений по породам:

- подавляющее преобладание какой-либо одной древесной породы (8 единиц состава и более),
- простое преобладание какой-либо одной древесной породы (5–7 единиц состава),
- смешанный состав по породам.

5. Группы возраста насаждений: молодняки 1 класса возраста; молодняки 2 класса возраста, средне возрастные и приспевающие насаждения; спелые и перестойные насаждения.

6. Группы полнот насаждений: низкополнотные (полноты 0,3–0,4); среднеполнотные (полноты 0,5–0,7); высокополнотные (полноты 0,8 и более).

7. Группы типов условий местопроизрастания насаждения (по влажности почв): свежие и влажные; сырые и мокрые.

8. Наличие и характеристики второго яруса насаждения (естественного или искусственного происхождения).

Отсутствует: еловый густой (полнота 0,6 и более); еловый редкий (полнота 0,5 и менее); дубовый густой (полнота 0,6 и более); дубовый редкий (полнота 0,5 и менее).

9. Древесный запас на 1га. (включая запас второго яруса) и единичных деревьев: до 60 м<sup>3</sup>; 61–130 м<sup>3</sup>; 131–220 м<sup>3</sup>; 221–330 м<sup>3</sup>; 331 м<sup>3</sup> и более.

10. Разновозрастные насаждения: не выражена; выражена (примесь старо-возрастной части насаждения более 25% от общего запаса).

11. Наличие сухостойных деревьев в насаждении: незначительное (10 м<sup>3</sup>/га и менее); повышенное (до 30% от запаса сырораствующей части насаждения); высокое (более 30% от запаса сырораствующей части насаждения).

При этом получают нижеследующие возможные постановки задач классификации лесохозяйственных объектов.

1. Разделение площади на категории земель (выбор между лесом – контуры 1–13, 15–25 и сенокосом – контуры 14 и 26).

2. Разделение покрытой лесом площади на участки с преобладанием разных породных групп – хвойные, твердолиственные и мягколиственные, а сенокосов – на суходольные (контур 14) и заболоченные (контур 26).

Ниже следует перечисление контуров, соответствующих породным группам для покрытой лесом площади.

Хвойные группы пород (ель) – контуры 6, 8, 10, 11, 21.

Твердолиственные группы пород (дуб) – контуры 4, 7.

Мягколиственные группы пород (береза, липа, осина, ольха черная) – контуры 1–3, 5, 9, 12–13, 15–20, 22–25.

3. Разделение площади на категории земель и по группам состава – по группам пород:

а) группа хвойных пород (контуры 6, 8, 10, 11, 21, которые могут подразделяться по п. 3.1 на контуры 6, 11, по п. 3.2 – на контуры 8, 21, а по п. 3.3 – на контур 10);

б) твердолиственные группы пород (контуры 4, 7, которые могут подразделяться по п. 3.2 на контур 4, а по п. 3.3 – на контур 7);

в) мягколиственные группы пород (контуры 1–3, 5, 9, 12, 13, 15–20, 22–25, которые могут подразделяться по п. 3.1 на контуры 1–3, 5, 9, 12, 13, 15–20, 23–25, а по п. 3.2 на контур 22).

Проведено испытание алгоритма при решении следующих задач классификации.

1. Разделение площади на категории земель (выбор между лесом – контуры 1–13, 15–25 и сенокосом – контуры 14 и 26).

Расчеты проведены для нижеследующих примеров.

1.1. Разделение площади на категории земель (выбор между лесом – контуры 1–13, 15–25 и сенокосом – контуры 14 и 26) при использовании четырех каналов (изображений): серого (GREY), красного (RED), зеленого (GREEN) и синего (BLUE); при использовании 16 – битовых изображений и 8 – битовых изображений. Радиус зрачка системы равен 2 пикселям при первом голосовании; повторное голосование проводилось в скользящем квадрате с размером стороны 13 пикселей. Для каждого класса с помощью схемы рис. 2 задано 5 объектов-эталонов.

1.2. Разделение площади на категории земель (выбор между лесом – контуры 1–13, 15–25 и сенокосом – контуры 14 и 26) при использовании семи каналов (изображений): серого (GREY), красного (RED), зеленого (GREEN), синего (BLUE), желтого (YELLOW) – в виде полусуммы красного и зеленого каналов (RED + GREEN), канала (RED + BLUE) в виде полусуммы этих каналов, канала (GREEN + BLUE) в виде полусуммы этих каналов при использовании 16-битовых изображений и 8-битовых изображений. Радиус зрачка системы равен 2 пикселям при первом голосовании; повторное голосование в скользящем квадрате с размером стороны 13 пикселей. Для каждого класса с помощью схемы рис. 2 задано 5 объектов – эталонов аналогично предыдущему расчету.

2. Разделение покрытой лесом площади на участки с преобладанием разных породных групп – хвойные, твердолиственные и мягколиственные (см. показатель 1), а сенокосов – на суходольные (контур 14) и заболоченные (контур 26).

Различаемые объекты:

1. Сенокос суходольный (контур 14).

2. Сенокос заболоченный (контур 26).

3. Хвойные группы пород – контуры 6, 8, 10, 11, 21.

4. Лиственные группы пород – контуры 1–5, 7, 9, 12–13, 15–20, 22–25.

Проведены следующие расчеты:

2.1. Разделение территории на 4 объекта: сенокос суходольный (контур 14), сенокос заболоченный (контур 26), хвойные группы пород – контуры 6, 8, 10, 11, 21, лиственные группы пород – контуры 1–5, 7, 9, 12–13, 15–20, 22–25. Используются семь каналов (изображений): серого (GREY), красного (RED), зеленого (GREEN), синего (BLUE), желтого (YELLOW) – в виде полу суммы красного и зеленого каналов (RED + GREEN), канала (RED + BLUE) в виде полусуммы этих каналов, канала (GREEN + BLUE) в виде полу суммы этих каналов при использовании 16-битовых изображений. Радиус зрачка системы равен 2 пикселям при первом голосовании; повторное голосование в скользящем квадрате с размером стороны 13 пикселей. Для каждого класса с помощью схемы рис. 2 задано 5 объектов–эталонов.

2.2. Разделение территории на 4 объекта: сенокос суходольный (контур 14), сенокос заболоченный (контур 26), хвойные группы пород – контуры 6, 8, 10, 11, 21, лиственные группы пород – контуры 1–5, 7, 9, 12–13, 15–20, 22–25. Используются семь каналов (изображений): серого (GREY), красного (RED), зеленого (GREEN), синего (BLUE), желтого (YELLOW) – в виде полу суммы красного и зеленого каналов (RED + GREEN), канала (RED + BLUE) в виде полусуммы этих каналов, канала (GREEN + BLUE) в виде полусуммы этих каналов; используются 8-битовые изображения. Радиус зрачка системы равен 2 пикселям при первом голосовании; повторное голосование в скользящем квадрате с размером стороны 13 пикселей. Для каждого класса с помощью схемы рис. 2 задано 5 объектов – эталонов аналогично предыдущему расчету .

3. Разделение площади на сенокосы и участки с преобладанием хвойных и лиственных породных групп (всего 3 объекта).

Различаемые объекты:

1. Сенокос суходольный (контур 14) и заболоченный (контур 26).

2. Хвойные группы пород – контуры 6, 8, 10, 11, 21.

3. Лиственные группы пород – контуры 1–5, 7, 9, 12–13, 15–20, 22–25.

Проведены следующие расчеты.



3.1. Разделение территории на 3 объекта: сенокос (контуры 14 и 26), хвойные группы пород – контуры 6, 8, 10, 11, 21, лиственные группы пород – контуры 1–5, 7, 9, 12–13, 15–20, 22–25. Используются семь каналов (изображений): серого (GREY), красного (RED), зеленого (GREEN), синего (BLUE), желтого (YELLOW) – в виде полу суммы красного и зеленого каналов (RED + GREEN), канала (RED + BLUE) в виде полусуммы этих каналов, канала (GREEN + BLUE) в виде полу суммы этих каналов; используются 16-битовые изображения. Радиус зрачка системы равен 2 пикселям при первом голосовании; повторное голосование в скользящем квадрате с размером стороны 13 пикселей. Для каждого класса с помощью схемы рис. 2 задано 5 объектов – эталонов аналогично предыдущему расчету.

Ниже приведены формулировки задач классификации, для которых надежное решение возможно при замене исходной задачи классификации с  $R$  классами на  $R$  задач классификации с двумя классами:  $i$ -й объект и остальные объекты.

4. Разделение площади на участки с преобладанием разных породных групп – хвойные, твердолиственные и мягколиственные, а сенокосов – на суходольные и заболоченные (всего 5 объектов)

Нумерация объектов:

1. Сенокос суходольный (контур 14).
2. Сенокос заболоченный (контур 26).
3. Хвойные группы пород – контуры 6, 8, 10, 11, 21.
4. Твердолиственные группы пород (дуб) – контуры 4, 7.
5. Мягколиственные группы пород (береза, липа, осина, ольха черная) – контуры 1–3, 5, 9, 12–13, 15–20, 22–25.

5. Разделение площади на категории земель и по группам состава пород.

Нумерация объектов:

1. Сенокос суходольный (контур 14).
2. Сенокос заболоченный (контур 26).
3. Подавляющее преобладание (8 единиц состава и более) хвойных пород (контуры 6, 11).
4. Простое преобладание (5–7 единиц состава) хвойных пород (контуры 8, 21).

5. Простое преобладание (5–7 единиц состава) дуба (контур 4).

6. Подавляющее преобладание (8 единиц состава и более) мягколиственных групп пород (контур 1–3, 5, 9, 12, 13, 15–20, 23–25).

7. Простое преобладание (5–7 единиц состава) мягколиственных групп пород (контур 22).

8. Смешанный состав по породам (контур 7, 10).

**Заключение.** Результаты расчетов позволяют сделать следующие выводы:

1. При разделении площади на категории земель (выбор между лесом – контуры 1–13, 15–25 и сенокосом – контуры 14 и 26) качество классификации при использовании 8-битовых и 16-битовых изображений (при четырех и семи каналах изображений) примерно одинаковое.

2. При разделении территории на 4 объекта: сенокос суходольный (контур 14), сенокос заболоченный (контур 26), хвойные группы пород – контуры 6, 8, 10, 11, 21, лиственные группы пород – контуры 1–5, 7, 9, 12–13, 15–20, 22–25 качество классификации при использовании 8-битовых и 16-битовых изображений заметно различается.

3. При сокращении количества различаемых объектов до трех в примере п.2 (разделение территории на 3 объекта: сенокос (контур 14 и 26), хвойные группы пород – контуры 6, 8, 10, 11, 21, лиственные группы пород – контуры 1–5, 7, 9, 12–13, 15–20, 22–25) результаты классификации становятся более правдоподобными. Поэтому можно сделать следующий вывод: для повышения качества классификации целесообразно заменять исходную задачу классификации с  $R$  классами на  $R$  задач классификации с двумя классами:  $i$ -й объект ( $i = 1, 2, \dots, R$ ) и остальные ( $R-1$ ) объекты, а также использовать повышенную размерность цветового вектора.

5. Рассмотренный выше метод классификации относится к методам классификации с помощью неадаптивных нейронных сетей и может использоваться для решения тех задач классификации, для которых не удается использовать адаптивные нейронные сети.

### Библиографический список

1. Терентьев И. В. О надежном распознавании объектов на аэрокосмических изображениях земной поверхности // Исследование Земли из Космоса. 1999 г., № 5, с. 57–64.

2. Терентьев И. В. О помехоустойчивости многоальтернативного дискретного накопителя // Радиотехника, т.24, № 7, 1969, с. 28–36.

3. Терентьев И. В. Построение надежной иерархической системы понятий для распознавания объектов на изображениях земной поверхности. (Тезисы доклада ) // Труды междунар. конф. «Пятьдесят лет развития кибернетики» 5–7 октября 1999 г., СПб.: Мин. обр. РФ, Совет по кибернетике РАН, Санкт-Петербург. н.-техн. общество приборостроителей им. С.И. Вавилова, Метрологич. Академия, С-Петербур. гос. техн. университет, 1999, с. 67–68.

4. О достоверном выделении лесохозяйственных объектов. Исследование Земли из Космоса // В. И. Березин, В. И. Воробьев, Н. П. Морус, Г.А. Васильев, И.В. Терентьев, М. А. Шубина/ 2003, № 4, с. 55–62.

5. Терентьев И. В. Различение отдельно стоящих рукописных символов с помощью методов оптимизации // Труды 6 междунар. н.-техн. конф. «Компьютерное моделирование 2005». 28 июня – 2 июля 2005 г., СПб.: Изд-во СПб гос. политехнического университета, 2005, с.407–408.

**Л. В. Уткин**, доктор технических наук, профессор,  
**Нгуен Ван Х.**, аспирант

## **ПЕССИМИСТИЧЕСКИЙ ПОДХОД К МНОГОКРИТЕРИАЛЬНОМУ ПРИНЯТИЮ РЕШЕНИЙ В РАМКАХ ТЕОРИИ СЛУЧАЙНЫХ МНОЖЕСТВ**

**Введение.** Метод анализа иерархий (МАИ) предназначен для решения многокритериальных задач принятия решений (МЗПР). Детальное описание метода можно найти в ряде работ, включая [1]. Достоинствами МАИ являются: универсальность, математическое обоснование и простота. Однако метод имеет ряд недостатков, к которым относятся необходимость построения большого числа матриц парных сравнений. МАИ дает только жесткие оценки альтернатив и не допускает неопределенности в суждениях. Кроме того, метод использует линейную свертку критериев, которая в ряде случаев может давать некорректные результаты.

Для того чтобы обойти приведенные выше недостатки, в литературе рассмотрено значительное число модификаций МАИ, направленных на ослабление слишком жестких требований, предъявляемых к экспертам.

Одна из таких модификаций основана на использовании теории Демпстера-Шейфера и использует неточные экспертные оценки в качестве исходных данных [3]. Этот метод сокращенно будем обозначать ДШ/МАИ (Демпстер–Шейфер / Метод анализа иерархий). В этой модификации вместо сравнения отдельных альтернатив между собой эксперту предлагается по каждому из критериев выделить из множества альтернатив подгруппы и для них экспертом определяется степени их предпочтения в заданной шкале по отношению к другим альтернативам.

Другой модификацией является метод, предложенный в работе [2], который также использует теорию Демпстера–Шейфера и обобщает метод ДШ/МАИ.

В предлагаемой работе рассматривается новый подход или модификация МАИ, который, с одной стороны, обобщает метод ДШ/МАИ и устраняют ряд его недостатков, а с другой стороны, рассматривает пессимистическую стратегию принятия решений или максиминный подход. При этом главной задачей в рамках предлагаемого подхода является поиск эффективных алгоритмов решения оптимизационных задач, которые возникают при реализации нового подхода.

### Элементы теории Демпстера-Шейфера

Пусть  $\Omega$  – некоторое множество, которое в теории свидетельств называется универсальным множеством. Предположим, что  $N$  наблюдений или измерений элемента  $\omega \in \Omega$  было получено в качестве информации об объекте, принимающем значения из  $\Omega$ . При этом предполагается, что результат измерений или наблюдений является неточным, т.е. представляет из себя некоторый интервал (подмножество)  $B$  значений  $\Omega$ . Пусть  $a_i$  означает количество наблюдаемых подмножеств  $B_i \subseteq \Omega$  и  $Po(\Omega)$  – множество всех подмножеств  $\Omega$  (множество мощности). Частотная функция  $m$ , называемая базовой вероятностью, определяется как [4]

$$m : Po(\Omega) \rightarrow [0,1], m(\emptyset) = 0, \sum_{B \in Po(\Omega)} m(B) = 1.$$

Базовая вероятность может быть получена следующим образом:

$$m(B_i) = a_i / N.$$

Функция доверия, обозначаемая  $\mathbf{Bel}(B)$ , и функция правдоподобия, обозначаемая  $\mathbf{Pl}(B)$ , события  $B \subseteq \Omega$  определяются как

$$\mathbf{Bel}(B) = \sum_{B_i : B_i \subseteq B} m(B_i), \quad \mathbf{Pl}(B) = \sum_{B_i : B_i \cap B \neq \emptyset} m(B_i).$$

Доверие  $\mathbf{Bel}(B)$  и правдоподобие  $\mathbf{Pl}(B)$  можно рассматривать как нижнюю и верхнюю вероятности события  $B$ , т.е.  $\mathbf{Bel}(B) \leq \Pr(B) \leq \mathbf{Pl}(B)$ .

### Оценки о критериях и альтернативах

Имеется множество альтернатив  $A = \{A_1, \dots, A_n\}$ , состоящее из  $n$  элементов, и множество критериев  $C = \{C_1, \dots, C_r\}$ , состоящее из  $r$  элементов. Эксперты выбирают некоторые подмножества  $B_k$  альтернатив из множества мощности  $Po(A)$  в соответствии с критерием  $C_j$  из  $C$ . Они также выбирают некоторые подмножества критериев  $D_i$  в качестве наиболее важных критериев. В соответствии с введенными обозначениями экспертные оценки представляются в виде предпочтений  $B_k \pm A$ . На основе полученных оценок вычисляются базовые вероятности подмножеств критериев  $m(D_i)$  и подмножеств альтернатив по заданным критериям  $m(B_i | C_j)$ .

Таблица 1

#### Экспертные оценки альтернатив

	$A_1$	$A_2$	$A_3$	$A_1A_2$	$A_1A_3$	$A_2A_3$	$A_1A_2A_3$
	$B_1$	$B_2$	$B_3$	$B_4$	$B_5$	$B_6$	$B_7$
$a_i^{(1)}$	3	2	2	3	0	0	0
$m(B_i   C_1)$	0.3	0.2	0.2	0.3	0	0	0
$\mathbf{Bel}_1(B_k)$	0.3	0.2	0.2	0.8	0.5	0.4	1
$\mathbf{Pl}_1(B_k)$	0.6	0.5	0.2	0.8	0.5	0.4	1
$a_i^{(2)}$	1	0	3	1	2	3	0
$m(B_i   C_2)$	0.1	0	0.3	0.1	0.2	0.3	0

$\text{Bel}_2(B_k)$	0.1	0	0.3	0.2	0.6	0.6	1
$\text{Pl}_2(B_k)$	0.4	0.4	0.8	0.7	1	0.9	1

Рассмотрим в качестве иллюстрации МЗПР, в которой ЛПР должен выбрать «наилучший» транспорт. Три альтернативы (ж/д –  $A_1$ , автомобильный –  $A_2$ , водный –  $A_3$  транспорт) оцениваются по двум критериям: надежность доставки ( $C_1$ ) и стоимость ( $C_2$ ). Пять экспертов выбрали  $D_1 = \{C_1\}$  в качестве наиболее важного критерия, два эксперта выбрали  $D_2 = \{C_2\}$  и три эксперта затруднились выбрать один критерий, т.е. они выбрали  $D_3 = \{C_1, C_2\}$ . Отсюда  $m(D_1) = 0.5$ ,  $m(D_2) = 0.2$  и  $m(D_3) = 0.3$ . Оценки для альтернатив даны в табл. 1, где также представлены рассчитанные значения функций доверия и правдоподобия в соответствии с каждым критерием (индексы 1 и 2).

Следующей задачей является комбинирование имеющихся оценок для получения обобщенных весов всех альтернатив.

### Критерий принятия решений

Метод для комбинирования и обработки неполной информации полностью зависит от критерия принятия решений. Грубо говоря, большая часть методов принятия решений состоит из комбинирования или свертки различных локальных критериев из  $S$  с вычислением функции  $F$ , называемой глобальным критерием, которую необходимо максимизировать. Глобальный критерий при конечном множестве альтернатив  $A$  может быть представлен следующим образом:

$$F(\mathbf{w}, \mathbf{u}_k) \rightarrow \max_A.$$

Здесь  $\mathbf{W} = (w_1, \dots, w_r)$  – вектор «весов» критериев;  $\mathbf{u}_k = (u_{1k}, \dots, u_{rk})$ ,  $k = 1, \dots, n$  – вектор полезностей  $k$ -й альтернативы в соответствии с каждым критерием из  $\{C_1, \dots, C_r\}$ . В частности, один из наиболее широко применяемых критериев – линейная функция, т.е.

$$F(\mathbf{w}, \mathbf{u}_k) = \sum_{i=1}^r (w_i \cdot u_{ik}).$$

Однако, несмотря на широкое использование линейной функции, она имеет ряд недостатков. Поэтому в случае пессимистического принятия решения, применяется максиминный метод, для которого

$$F(\mathbf{w}, \mathbf{u}_k) = \min_{i=1, \dots, r} (w_i \cdot u_{ik}).$$

При наличии только ограниченной информации о векторах  $\mathbf{w}$  и  $\mathbf{u}_k$  можно получить только ограниченную информацию о функции  $F$ . Пусть вектора  $\mathbf{w}$  и  $\mathbf{u}_k$  принимают значения из множеств  $W$  и  $U_k$  соответственно. Здесь  $U_k$  – произведение  $r$  интервалов  $U_{ik}$ . Тогда можно сказать, что функция  $F$  принадлежит интервалу или множеству интервалов. Если все ограничения для  $\mathbf{w}$  линейны, то множество  $W$  является выпуклым. Используя свойство монотонности функции  $F$ , можно достаточно просто доказать, что ее значения принадлежат только одному интервалу  $[F_1(k), F_2(k)]$ .

Следующий вопрос – как использовать критерий при наличии множества возможных значений функций  $F$ . Выбор «наилучшей» альтернативы может быть основан на сравнении интервалов функции  $F$ . Существует большое количество методов сравнения. Здесь предлагается использовать наиболее обоснованный метод, основанный на параметре осторожности или пессимизма  $\eta$  д  $[0, 1]$ , имеющего такой же смысл, как и параметр оптимизма в критерии Гурвица. В соответствии с этим методом «наилучшая» альтернатива выбирается так, что линейная комбинация  $\eta F_1 + (1 - \eta) F_2$  достигает своего максимума.

Следующая задача – разработка алгоритма вычисления нижней и верхней границ функции  $F$  при использовании максиминного метода.

### **Алгоритм вычисления нижней и верхней границ**

Если предположить, что веса альтернатив представляются при помощи функций доверия и правдоподобия, то можно записать

$$F_1(k) = \mathbf{Bel}(B_k) = \inf_{\mathbf{w} \in W} \min_{j=1, \dots, r} (w_j \cdot \mathbf{Bel}_j(B_k)),$$

$$F_2(k) = \mathbf{Pl}(B_k) = \sup_{\mathbf{w} \in W} \min_{j=1, \dots, r} (w_j \cdot \mathbf{Pl}_j(B_k)).$$

Здесь  $W$  – множество векторов  $w$ , определенное имеющейся информацией о критериях.

Вектора  $w$  в  $W$  эквивалентны. Отсюда следует, что произвольный интервал  $[\mathbf{Bel}^*(B_k), \mathbf{Pl}^*(B_k)]$  при некотором  $w \in W$  лежит в интервале  $[\mathbf{Bel}(B_k), \mathbf{Pl}(B_k)]$ . Поэтому вычисляется наибольший интервал. При наличии базовых вероятностей  $m(D_k)$  подмножеств  $D_k$ , функции доверия и правдоподобия подмножества  $D_k$  вычисляются как

$$\mathbf{Bel}(D_k) = \sum_{i: D_i \subseteq D_k} m(D_i),$$

$$\mathbf{Pl}(D_k) = \sum_{i: D_i \cap D_k \neq \emptyset} m(D_i), \quad k = 1, \dots, 2^r - 1.$$

Предположим, что  $j$ -й критерий выбирается экспертами с некоторой неизвестной вероятностью  $p_j$ , так что выполняется условие  $\sum_{j=1}^r p_j = 1$ . Тогда вероятности критериев удовлетворяют системе неравенств

$$\mathbf{Bel}(D_k) \leq \sum_{j: C_j \in D_k} p_j \leq \mathbf{Pl}(D_k), \quad k = 1, \dots, 2^r - 1. \quad (1)$$

Множество неравенств образует множество  $P$  возможных распределений  $p = (p_1, \dots, p_r)$ . Функции доверия и правдоподобия подмножеств альтернатив в соответствии с  $j$ -м критерием имеют вид

$$\mathbf{Bel}_j(B_k) = \sum_{i: B_i \subseteq B_k} m(B_i | C_j), \quad \mathbf{Pl}_j(B_k) = \sum_{i: B_i \cap B_k \neq \emptyset} m(B_i | C_j).$$

Зафиксируем  $p$  из  $P$ . Применяя максиминный метод, получаем

$$\mathbf{Bel}_p(B_k) = \min_{j=1, \dots, r} (p_j \cdot \mathbf{Bel}_j(B_k)), \quad \mathbf{Pl}_p(B_k) = \min_{j=1, \dots, r} (p_j \cdot \mathbf{Pl}_j(B_k)).$$

Полученные функции доверия и правдоподобия зависят от  $p$ . Следовательно, найдем нижнюю функцию доверия и верхнюю функцию правдоподобия, решив задачи оптимизации:

$$\mathbf{Bel}(B_k) = \inf_{p \in P} \mathbf{Bel}_p(B_k) = \inf_{p \in P} \min_{j=1, \dots, r} (p_j \cdot \mathbf{Bel}_j(B_k)), \quad (2)$$



$$\mathbf{Pl}(B_k) = \sup_{p \in P} \mathbf{Pl}_p(B_k) = \sup_{p \in P} \min_{j=1, \dots, r} (p_j \cdot \mathbf{Pl}_j(B_k)) \quad (3)$$

при ограничении  $\sum_{j=1}^r p_j = 1$  и (1).

Фактически мы используем здесь функцию  $F(\mathbf{w}, \mathbf{u}_k)$  весов  $w_1 = p_1, \dots, w_r = p_r$  критериев  $C_1, \dots, C_r$ . Доверие  $\mathbf{Bel}_j(B_k)$  и правдоподобие  $\mathbf{Pl}_j(B_k)$  являются нижними и верхними границами значение  $u_{jk}$ . Используя свойства функции  $F$ , описанные выше, можно утверждать, что интервал  $[\mathbf{Bel}(B_k), \mathbf{Pl}(B_k)]$  определяет границы  $[F_1(k), F_2(k)]$ .

Первая задача

Рассмотрим задачу (2). Это – задача нелинейной оптимизации. Введем новую переменную  $G = \min_{j=1, \dots, r} (p_j \cdot \mathbf{Bel}_j(B_k))$ . Тогда задачу (2) можно переписать в виде

$$\mathbf{Bel}(B_k) = \inf_{p \in P} G$$

при ограничениях  $\sum_{j=1}^r p_j = 1$ , (1) и

$$G \leq p_j \cdot \mathbf{Bel}_j(B_k), \quad j = 1, \dots, r.$$

Получена задача линейного программирования, имеющая  $r+1$  переменных. Однако она не имеет решения, так как при минимизации целевой функции, переменная  $G$  неограниченно убывает. Как ограничить эту переменную? Из определения  $G$  следует, что ее оптимальное значение есть  $p_j \cdot \mathbf{Bel}_j(B_k)$ . Тогда оптимальное решение соответствует равенству в одном из ограничений  $G \leq p_j \cdot \mathbf{Bel}_j(B_k)$ , которое ограничивает  $G$ . Следовательно, необходимо решить таких  $r$  линейных задач, что  $i$ -я задача имеет одно ограничение  $G = p_i \cdot \mathbf{Bel}_i(B_k)$  вместо  $G \leq p_i \cdot \mathbf{Bel}_i(B_k)$ . Пусть  $i$ -я задача имеет оптимальное решение  $G^{(i)}$ . Тогда

$$\mathbf{Bel}(B_k) = \min_{i=1, \dots, r} G^{(i)}.$$

Отметим, что  $i$ -я задача является линейной. Поэтому ее решение можно найти на множестве крайних точек  $\mathbf{extr}(P)$  выпуклого многогранника

Р распределений, образованного линейными ограничениями. Процедура вычисления крайних точек хорошо известна. Так как крайние точки не зависят от целевой функции, то нижняя функция доверия вычисляется подстановкой крайних точек  $\mathbf{extr}(P)$  в ограничения  $r$  линейных задач. В результате имеем  $r \cdot M$  задач программирования, где  $M$  – число крайних точек.

Наименьшее оптимальное значение  $G$  для  $r \cdot M$  задач является оптимальным решением задачи (2). Следовательно, задача (2) может быть переписана как

$$\mathbf{Bel}(B_k) = \min_{i=1, \dots, M} \min_{j=1, \dots, r} (p_j^{(i)} \cdot \mathbf{Bel}_j(B_k)),$$

где  $p^{(i)} = (p_1^{(i)}, \dots, p_r^{(i)})$  –  $i$ -я крайняя точка.

Вторая задача

Рассмотрим задачу (3). Это – также задача нелинейной оптимизации. Используя новую переменную  $G = \min_{j=1, \dots, r} (p_j \cdot \mathbf{Pl}_j(B_k))$ , задачу (3) переписем как

$$\mathbf{Pl}(B_k) = \sup_{p \in P} G$$

при ограничениях  $\sum_{j=1}^r p_j = 1$ , (1) и  $G \leq p_j \cdot \mathbf{Pl}_j(B_k)$ ,  $j = 1, \dots, r$ .

Предположим, что информация о критериях отсутствует. Тогда многогранник  $P$  образован только одним ограничением  $\sum_{j=1}^r p_j = 1$  и имеет крайние точки

$$(1, 0, \dots, 0), (0, 1, \dots, 0), \dots, (0, 0, \dots, 1).$$

Из (2) и (3) следует в этом случае, что  $\mathbf{Bel}(B_k) = 0$  и  $\mathbf{Pl}(B_k) = \max_{j=1, \dots, r} \mathbf{Pl}_j(B_k)$ .

Числовой пример

Вернемся к примеру о выборе «наилучшего» транспорта. Функции доверия и правдоподобия подмножеств  $D_1, D_2, D_3$  равны

$$\mathbf{Bel}(D_1) = m(D_1) = 0.5, \quad \mathbf{Pl}(D_1) = m(D_1) + m(D_3) = 0.8,$$

$$\mathbf{Bel}(D_2) = m(D_2) = 0.2, \quad \mathbf{Pl}(D_2) = m(D_2) + m(D_3) = 0.5,$$

$$\mathbf{Bel}(D_3) = \mathbf{Pl}(D_3) = 1.$$

Вычислим функции доверия и правдоподобия альтернатив  $A_1, A_2, A_3$ . Множество  $P$  распределений  $p = (p_1, p_2)$  имеет две крайние точки  $V = (0.8, 0.2)$ ,  $W = (0.5, 0.5)$ . Отсюда

$$\mathbf{Bel}(B_k) = \min \left\{ \min(0.8 \cdot \mathbf{Bel}_1(B_k), 0.2 \cdot \mathbf{Bel}_2(B_k)), \min(0.5 \cdot \mathbf{Bel}_1(B_k), 0.5 \cdot \mathbf{Bel}_2(B_k)) \right\}.$$

Например, рассматривая  $B_1 = \{A_1\}$ , можно записать (см. табл. 1)

$$\begin{aligned} \mathbf{Bel}(B_1) &= \min \left\{ \min(0.8 \cdot 0.3, 0.2 \cdot 0.1), \min(0.5 \cdot 0.3, 0.5 \cdot 0.1) \right\} = \\ &= \min(0.02, 0.05) = 0.02. \end{aligned}$$

Задача линейного программирования для вычисления функции правдоподобия  $A_1$  имеет вид

$$\mathbf{Pl}(B_1) = \mathbf{Pl}(A_1) = \sup_{p \in P} G$$

при ограничениях  $p_1 + p_2 = 1$  и

$$0.5 \leq p_1 \leq 0.8, \quad 0.2 \leq p_2 \leq 0.5, \quad G \leq p_1 \cdot 0.6, \quad G \leq p_2 \cdot 0.4.$$

Оптимальное решение –  $G = \mathbf{Pl}(A_1) = 0.2$ . Функции доверия и правдоподобия других альтернатив вычисляются аналогично:  $\mathbf{Bel}(A_2) = 0$ ,  $\mathbf{Pl}(A_2) = 0.2$ ,  $\mathbf{Bel}(A_3) = 0.06$ ,  $\mathbf{Pl}(A_3) = 0.16$ . Таким образом, имеем три интервала  $[0.02, 0.2]$ ,  $[0, 0.2]$ ,  $[0.06, 0.16]$ . Принимая  $\eta = 0.6$ , получаем три точных значения  $0.092$ ,  $0.08$ ,  $0.1$ , соответствующих трем альтернативам. Отсюда можно сделать вывод, что третья альтернатива является оптимальной.

**Заключение.** Предлагаемый в работе подход к пессимистическому МЗПР позволяет использовать неточные и групповые оценки альтернатив и критериев. С этой точки зрения этот подход можно рассматривать, как обобщение метода ДШ/МАИ. Задачи оптимизации, решение которых необходимо получить для поиска оптимальной альтернативы, являются линейными и могут быть решены известными методами, например сим-

плекс методом. Это делает реализацию подхода достаточно простой с вычислительной точки зрения.

Необходимо также отметить, что предлагаемый подход достаточно просто обобщается на случай предоставления экспертами дополнительной числовой информации в виде оценок по некоторой числовой шкале. Это также обобщает подход и делает его достаточно универсальным.

### Библиографический список

1. Саати Т. Принятие решений. Метод анализа иерархий. Радио и связь. Москва. 1993.
2. Уткин Л.В., Симанова Н.В. Метод анализа иерархий при неполной информации о критериях и альтернативах // Нечеткие системы и мягкие вычисления. т. 2, № 2, с. 31-40. 2007.
3. Beynon M., Curry B., Morgan P. The Dempster-Shafer theory of evidence: an alternative approach to multicriteria decision modeling // Omega. 28, pp. 37–50. 2000.
4. Dempster A.P. Upper and lower probabilities induced by a multi-valued mapping // *Annales of Mathematical Statistics*, 38, pp. 325–339. 1967.

**Н. П. Васильев**, кандидат технических наук, доцент  
**А. И. Ковешников**, ведущий программист  
**В. А. Пресняков**, кандидат физико-математических наук, доцент

## **ВАРИАНТ РЕШЕНИЯ ЗАДАЧИ ЭКСПОРТА ДАННЫХ**

**Введение.** Задача экспорта данных из одной информационной системы в другую зачастую является камнем преткновения, как для пользователей, так и для разработчиков. Как правило, решение этой задачи требует больших затрат, связанных с ручной набивкой экспортируемых данных, а затем с выявлением и исправлением неизбежных ошибок оператора. В худшем случае решение просто не представляется возможным по тем или иным причинам. Например, разработчики системы не предусмотрели каких-либо вариантов экспорта, либо простая набивка данных является слишком объемной, и по этой причине, практически не выполнимой.

В настоящей статье выявляются иные возможные трудности, связанные с экспортом данных, носящие принципиальный характер, и предлагается универсальный подход к их преодолению. Предлагаемые приемы реализованы и опробованы в автоматизированной информационной системе Alpine, разработчиками которой являются авторы. Подход применим к любой реляционной базе данных под управлением SQL-сервера. Так данные в Alpine хранятся в базе под управлением MS SQL [1-2].

Следует отметить, что клиентская часть, поставляемая вместе с SQL-серверами, обычно позволяет экспортировать таблицы в том или ином формате, но для целей импорта данных в иные системы, как правило, эта возможность является бесполезной.

**Постановка задачи.** Экспорт данных из одной информационной системы, очевидно, предполагает наличие другой информационной системы, в которую эти данные будут импортироваться. Условно обозначим первую систему через А, а вторую через В. Чтобы задача имела смысл, требуется выполнение ряда условий.

Во-первых, как система А, так и система В, должны хранить данные об одних и тех же объектах (сущностях), то есть, фактически, предназначаться для хранения одних и тех же данных (конечно, только в части, подлежащей экспорту, импорту). Вместе с тем, форма их представления в разных системах может быть и, как правило, бывает различна. Данные могут быть представлены различными по структуре таблицами, в полях которых они могут кодироваться по-разному. Более того, система В, в принципе, может и не являться реляционной.

Во-вторых, в системе В должен быть предусмотрен импорт, или загрузка тех, или иных данных, представленных в той, или иной форме. Как правило, это текстовый файл, в строках которого предусмотрены позиции для хранения атрибутов, подлежащих загрузке.

Таким образом, задача экспорта данных из системы А в систему В сводится к выборке этих данных в системе А и к представлению их в форме, пригодной для загрузки в системе В. Эта задача обычно существенно усложняется различными способами формирования ключевых колонок в системе А и в системе В, что неизбежно приводит к различным кодам, которые приписываются одним и тем же сущностям в системе А, и в системе В.

Поясним сказанное на простом примере. Пусть в системе А хранятся сведения о клиентах в таблице client:

Код клиента (codeA)	Наименование клиента (name)	Др. атрибуты клиентов (attr)
A1	Клиент 1	...
A2	Клиент 2	...
A3	Клиент 3	...
...	...	...

В системе В эти же клиенты, скорее всего, будут кодироваться иначе:

Код клиента (codeB)	Наименование клиента (name)	Др. атрибуты клиентов (attr)
B1	Клиент2	...
B2	Клиент1	...
B3	Клиент3	...

...	...	...
-----	-----	-----

Пусть в системе А фиксируются заказы, сделанные клиентами, в таблице order:

Код клиента (client)	Атрибуты заказа (attr)
A1	...
A2	...
A1	...
A3	...

Требуется экспортировать данные о заказах в систему В. Конечно, предполагается, что система В способна загружать эти данные, представленные, например, в обычном текстовом файле. Очевидно, экспорт таблицы с заказами из системы А для последующего ее импорта в систему В осложняется различными кодами клиентов в этих системах. Одни и те же клиенты в системе А имеют коды А1, А2, А3, а в системе В их коды соответственно В2, В1, В3.

Конечно, в реальных системах экспортируемые данные могут быть разбросаны по нескольким таблицам, а по-разному закодированы не только клиенты, но суть задачи при этом не изменяется.

**Примитивный вариант решения задачи.** Это подход предполагает создание вспомогательной таблицы, позволяющей сопоставлять кодам сущностей в системе А коды, которые приписываются этим же сущностям в системе В. В рассматриваемом примере такая таблица позволяет перекодировать клиентов под нужды системы В, в которую осуществляется импорт данных о заказах. Эта таблица (назовем ее client\_aux) имеет следующую структуру:

Код клиента в системе А (codeA)	Код клиента в системе В (codeB)
A1	B2
A2	B1
A3	B3

Если такая таблица создана и заполнена корректными значениями, то экспортируемые данные можно собрать с помощью следующего SQL запроса:

```
SELECT client_aux.codeB, order.attr  
FROM order, client_aux  
WHERE order.client=client_aux.codeA
```

Эти данные остается отформатировать и представить, например, в текстовом файле, который и будет служить для загрузки данных в систему В. В реальных ситуациях, как уже упоминалось, может потребоваться несколько таблиц перекодировки, которые могут иметь более сложную структуру, а данные придется собирать по нескольким таблицам, но, в любом случае, составить соответствующий SQL запрос не составит труда.

Очевидно, такой подход потребует от разработчиков программирования в клиентской части системы средств заполнения и корректировки вспомогательной таблицы `client_aux`, а также реализации проверки целостности этой таблицы. Проверка целостности в данном случае сводится к проверке наличия в таблице `client_aux` для каждого кода клиента в системе А, упомянутого в таблице `order`, соответствующего кода клиента в системе В.

Очевидно, такой подход не является универсальным, и если вдруг возникнет необходимость перекодировки других сущностей (например, в связи с новой задачей экспорта), то это опять потребует создания соответствующей таблицы и программирования средств ее заполнения и корректировки. Наоборот, если необходимость в экспорте каких-либо данных отпадает, то весь созданный для этой цели ассортимент средств будет являться ненужным хламом, от которого, как показывает практика, не так просто избавиться.

Кроме этого, описанный подход обладает еще одним недостатком, который не сразу бросается в глаза. Так в приведенном примере экспорта данных о заказах, очевидно, выполнять перекодировку следует только для тех клиентов, которые упоминаются в таблице с заказами `order`. Вместе с тем средств выявления таких клиентов не предусматривается. Выход остается один – перекодировать всех клиентов. Выполнение последней задачи может оказаться слишком объемным и в то же время не нужным.



Наконец, речи пока не идет о каком-либо универсальном механизме, позволяющем отработать SQL запрос экспорта данных и должным образом отформатировать эти данные в выходном файле.

**Универсальный вариант решения задачи.** Во-первых, попытаемся найти универсальное решение задачи перекодировки сущностей, то есть универсальный механизм, который позволяет кодам сущностей в системе А сопоставлять коды этих же сущностей в системе В. Такое сопоставление будем называть справочником конвертации. Оказывается, все справочники конвертации, которые могут потребоваться для экспорта данных, можно полностью описать с помощью трех таблиц.

Каждый такой справочник будет определяться, прежде всего, некоторым правилом выбора значений кодов из системы А, которым надо будет приписать соответствующие значения в системе В. Коды в системе А будем называть входными значениями справочника, а коды в системе В – выходными. Правило выбора входных значений можно представить некоторым SQL запросом. Так в нашем примере этот запрос довольно простой и выглядит следующим образом:

```
SELECT distinct client  
FROM order
```

Следует отметить, что входных полей в справочнике может быть в принципе несколько, и на практике такое случается часто. То же касается и выходных полей. Описываемый подход вполне подходит и для таких ситуаций.

Каждому справочнику требуется дать некоторое уникальное мнемоническое имя и какое-то название. Таким образом, первая таблица для целей универсального описания справочников конвертации (дадим ей имя `cnvrt_guide`) будет содержать следующие поля:

Идентификатор справочника (gdid)	Имя справочника (gdnm)	Правило выбора входных значений (tbl)	Название справочника (gdttl)
----------------------------------	------------------------	---------------------------------------	------------------------------

1	client_aux	SELECT distinct client FROM order	Справочник перекодировки клиентов
2	...	...	...

Для описания входных и выходных полей справочников может служить таблица со следующей структурой:

Идентификатор справочника (gdid)	Порядковый номер поля в справочнике (ord)	Тип поля (type)	Имя поля (fldnm)	Название поля (fldttl)
1	1	Входное поле	codeA	Код клиента в системе А
1	2	Выходное поле	codeB	Код клиента в системе В
2	...	...	...	...

Этой таблице дадим имя `snvrt_key`.

Наконец, в третьей таблице будут храниться значения входных и выходных полей введенные в справочник. Эта таблица (назовем ее `snvrt_val`) будет выглядеть так:

Идентификатор справочника (gdid)	Порядковый номер поля в справочнике (ord)	Строка справочника (row)	Значение поля (val)
1	1	1	A1
1	2	1	B2
1	1	2	A2
1	2	2	B1
1	1	3	A3
1	2	3	B3
2	...	...	...

Все три таблицы связаны полем `gdid`. В таблице `cnvrt_guide` это поле идентифицирует тот или иной справочник и является ключевым, в таблицах `cnvrt_key` и `cnvrt_val` это поле, очевидно, является внешним ключом [1].

Поля `ord` и `row` определяют колонку и строку в справочнике конвертации, то есть клетку, которая содержит значение, занесенное в поле `val` таблицы `cnvrt_val`.

При этом в таблицу `cnvrt_val` значения, соответствующие некоторой строке справочника конвертации, заносятся только тогда, когда эта строка будет заполнена какими-либо выходными значениями. Если же выходные значения не были введены, то отдельно только входные значения в этой таблице размещаться не должны. Таким образом, если справочник конвертации пустой, то есть не содержит выходных значений, то и в таблице `cnvrt_val` не будет строчек, соответствующих этому справочнику. Вот это собственно все простые правила, касающиеся заполнения представленных таблиц.

Вопрос теперь состоит в том, как этими таблицами пользоваться, то есть каким образом получить представление того или иного справочника конвертации. С этой целью вспомним, что первая половина справочника конвертации, содержащая значения входных полей, фактически, определяется SQL запросом, содержащимся в поле `tbl` таблицы `cnvrt_guide`. Эту половину требуется «сшить» со второй половиной справочника, содержащейся в таблице `cnvrt_val`.

Строки справочника, заданного своим идентификатором, содержащиеся в таблице `cnvrt_val`, то есть заполненные выходными значениями, можно представить следующим образом:

```
SELECT codeA.val as codeA, codeB.val as codeB  
FROM cnvrt_val codeA, cnvrt_val codeB  
WHERE  
codeA.row=codeB.row //все относится к одной и той же строке  
and codeA.ord=1 and codeB.ord=2 //поля по порядку их следования в  
справочнике  
and codeA.gdid=1 and codeB.gdid=1 //рассматриваем первый справочник
```

Это представление построено для нашего примера, то есть, для первого справочника (gdid=1). Очевидно, легко построить подобное представление любого справочника. Достаточно выбрать из таблицы `cnvrt_key` по порядку (по возрастанию значений в поле `ord`) все поля, относящиеся к справочнику с заданным идентификатором `gdid`, а затем для каждого поля сформировать соответствующие строки в утверждениях `SELECT`, `FROM` и `WHERE`. В результате, получим представление только тех строк справочника конвертации, которые были заполнены хотя бы одним выходным значением. Это представление в дальнейшем будем обозначать через `outselect`.

Как уже упоминалось, для получения полной картины остается полученное представление «сшить» с представлением входных значений, определяемых запросом в поле `tbl` таблицы `cnvrt_guide`, который для краткости обозначим через `userselect`. Сделать «сшивку» для нашего справочника можно так:

```
SELECT tbl.codeA, out.codeB  
FROM (userselect) tbl  
LEFT JOIN  
(outselect) out  
ON (out.codeA = tbl.codeA)
```

Сшивка осуществляется по входным значениям двух представлений `userselect` и `outselect`.

Подобные представления, очевидно, легко построить для любых справочников, с любым количеством входных полей. При этом для каждого входного поля в утверждении `ON` достаточно добавить соответствующее условие «сшивки». В данном примере условие «сшивки» приведено для наглядности в упрощенном варианте. Дело в том, что, входные поля в представлении `userselect`, скорее всего, будут отличаться по типу от этих же полей в представлении `outselect`. Если это так, то в утверждении `ON`, очевидно, потребуется дополнительное преобразование типов полей перед сравнением. Поскольку тип поля `val` в таблице `cnvrt_val` для любого справочника конвертации один и тот же, то легче всего организовать преобразование полей из представления `userselect` именно к этому типу.

Для нашей задачи экспорта данных о заказах достаточно создать представление справочника конвертации кодов клиентов с прежним именем `client_aux`:

```
CREATE VIEW client_aux AS  
SELECT tbl.codeA, out.codeB  
FROM (userselect) tbl  
LEFT JOIN  
(outselect) out  
ON (out.codeA = tbl.codeA)
```

Это представление является полноправной заменой одноименной таблицы преобразования кодов в описанном выше случае примитивного решения задачи, поэтому для экспорта данных о заказах можно использовать тот же самый SQL-запрос (см. выше).

Наконец, следует отметить, что клиентская часть системы, предназначенная для ввода и удаления справочников конвертации (то есть фактически для редактирования описанных выше трех таблиц), может также отслеживать создание и удаление соответствующих пользовательских представлений этих справочников.

**Заключение.** Экспорт данных из одной информационной системы в другую, как правило, принципиально осложняется необходимостью перекодировки сущностей. Сделать это без вмешательства разработчиков системы обычно не удастся, поскольку упрощенное решение этой задачи требует создания и поддержки со стороны серверной и клиентской частей системы соответствующих таблиц перекодировки. Для каждой задачи экспорта подобные таблицы являются уникальными. Со временем эти таблицы, когда необходимость в них отпадает, могут серьезно захлестить систему, если при этом еще учесть сопутствующие программные разработки на клиентской и серверной стороне.

Предложенное универсальное решение задачи позволяет создавать и удалять при необходимости любые представления, связанные с перекодировкой объектов. При этом оказывается можно обойтись всего лишь тремя таблицами, которые фактически решают любые проблемы, связанные с перекодировкой. Это решение, таким образом, свободно от указанных не-

достатков, с одной стороны, а с другой стороны, позволяет автоматически отслеживать только те входные коды, которые только и требуются для экспорта. Единое решение, таким образом, позволяет повысить информационную целостность системы, увеличить сроки ее существования.

Решение задачи успешно опробовано авторами в реальной автоматизированной системе Alpine, информационная часть которой реализована на базе сервера MSSQL. Принципиальных ограничений, связанных с использованием описанного подхода в системах с другими SQL-серверами, практически нет. Единственное требование к SQL-серверу – это возможность использования встроенных запросов [1].

#### **Библиографический список**

1. Джеймс Р. Гофф, Пол Н. Вайнберг SQL: Полное руководство. Киев: BHV, 2000. 606 с.
2. Тихомиров Ю.В. Microsoft SQL Server 7.0. СПб.: БХВ-Санкт Петербург, 1999. 720 с.

**И. В. Ганичев**, кандидат технических наук,  
**Е. Г. Троицкая, А. Г. Троицкая**, студенты

## **СИСТЕМНЫЙ АНАЛИЗ ИЕРАРХИЧЕСКОЙ МОДЕЛИ ЭВМ**

**Введение.** Задача анализа концепций построения и формирование полной многоуровневой иерархической модели ЭВМ с выделением всех конфликтных областей и узловых точек противоречий актуальна. Наглядность модели дает возможность обратить внимание на комплекс проблем в согласовании принципов верхних уровней иерархии и реализаций подсистем и компонентов нижних уровней, принять ряд мер для повышения эффективности и качества функционирования информационной системы.

Несоответствие принципов построения различных подсистем ЭВМ принято называть в современной литературе семантическим разрывом [1].

Настоящая работа имеет выраженную учебно-методическую направленность. Иерархическая модель ЭВМ может быть использована прежде всего в дисциплинах Теория информационных процессов и систем, Архитектура ЭВМ, Операционные системы и в разделах Информатики.

Цели работы:

- разработать и исследовать иерархическую цепочку моделей ЭВМ, соответствующую максимально широкому применению подхода стратификации,
- соединить аппаратную, программную и алгоритмическую составляющие (Hard, Soft, Brain) в единой модели ЭВМ с соотнесением их к соответствующим уровням иерархии,
- показать точки семантического разрыва в описании и функционировании ЭВМ (среди всех страт единой модели, модулей и подсистем ЭВМ и компонентов программного обеспечения).

Ставится задача построения единой многоуровневой иерархической модели ЭВМ, на которой можно показать наибольшее число источников противоречий, обозначающих семантические разрывы

Набор (или спектр) базовых моделей в иерархии при полноценном анализе должен быть достаточно широким: должен охватывать все представления о системе – от «философского или теоретико-познавательного описания ее замысла» до «материального воплощения», учитывать весь ряд «проектных представлений» и представлений «на языке выбранной научной теории» [2]. К сожалению, такой развернутой модели для ЭВМ не найдено в открытой литературе.

М. Месарович выдвинул плодотворную идею представления систем различного назначения в виде страт. Упомянутая в основополагающей работе [3] как пример модель ЭВМ содержит только две важные страты, ввиду слишком большой общности для нашего исследования интереса не представляет.

Совокупность страт представляет собой иерархическое многоуровневое описание системы, причем объяснение назначения системы с помощью элементов верхней страты по существу есть отражение и сжатое описание теоретико-познавательного замысла нижних страт системы. Каждая страта,

исходя из этой концепции, открывает путь к синтезу объекта на нижней страте.

Следуя автору [3], сформулируем главную отличительную идею стратифицированного, а значит и более независимого описания уровней иерархии:

*— при переходе от одной страты к другой, чем ниже мы спускаемся по иерархии, тем более детальным становится раскрытие системы, чем выше мы поднимаемся, тем яснее становится смысл и значение всей системы.*

Подъем вверх по стратам приводит в итоге к уяснению общих, потенциальных возможностей системы, привлечение элементов нижележащих (и более детализированных страт) дает более полное понимание функционирования реализации конкретной системы.

Данный подход в стратификации моделей наряду с ясным пониманием, в каких областях иерархии зарождается семантический разрыв, позволит выявить возможные актуальные направления развития архитектуры компьютеров.

Безусловно, успех стратификации описания зависит от целей его формирования, интересов исследователя, опыта и его квалификации. Агрегируя известное описание страт системы Ю. Черняка применительно к технике можно сформулировать следующую цепочку «Философия системы – Языковое описание и Проекты – Физическое воплощение».

Всякая система предназначена для решения конкретной проблемы и представляет собой целенаправленное органичное образование из множества элементов, подсистем и компонентов. В процессе взаимодействия частей системы проявляется принципиально новое качество, или свойство, которым не обладает ни одна из составляющих частей – целостность всей системы.

Известно из практики, что фактически многие процессы, реализуемые в аппаратуре, можно представить, моделируя программно работу части устройств, и обратно, программные решения – реализовать в аппаратуре. В этом проявляется дуальность проектирования.



Недостатки реализации одной стороны дуального аппаратно-программного комплекса обработки информации можно в определенной мере компенсировать доработкой альтернативных решений другой стороны.

Таким образом, стратифицированная модель ЭВМ для наших исследований также должна явно нести черты дуальности, возможности развития любой из двух ветвей реализации в зависимости от задач, которые ставятся перед проектом системы, а также критерия эффективности ее функционирования.

### **1. Стратифицированная модель**

Стратификация моделей – это эффективная и рациональная концепция формирования описания сложной системы с большей независимостью уровней иерархии. В связи с этим уровни абстрагирования, включающие стратифицированное описание, особо называют стратами. В конечном итоге они позволяют полнее выразить особенности системы и ее потенциальные возможности.

Перед нами стояла задача выявить страту, которая несет признаки дуальности, и так сформировать иерархию, чтобы наглядно была видна взаимосвязь и единство важных программно-аппаратурных компонентов системы.

В качестве страты, которая порождает сразу оба направления – аппаратное и программно-алгоритмическое может быть взято описание машины Тьюринга (МТ). В теоретической части курса «Информатики» студенты специальности 071900 знакомятся с математическим определением Тьюринга понятия алгоритма (по сути, уточнением) в форме абстрактного автомата, называемого машиной Тьюринга.

Совместно с инженером кафедры Т.В. Зубовской еще в 1997 г. на кафедре была поставлена лабораторная работа «Машина Тьюринга». МТ – абстрактный объект. Реализация имитатора МТ в среде Excel выявила аналогии составных частей с модулями ЭВМ: лента – память, головка чтения и записи – устройство ввода-вывода данных, конечный автомат – устройство управления. МТ не является реально действующим устройством, однако описание ее предельно конструктивно. В силу этого МТ постоянно использовалась в качестве основной модели для выяснения существа понятий «вычислительный процесс», «алгоритм», а также для выяснения связи между алгоритмом и вычислительными машинами.

МТ считают не только способом разрешения важных научных алгоритмических проблем, но и своеобразной моделью ЭВМ, что вполне соответствует нашим представлениям о верхней страте иерархической модели ЭВМ. МТ дает возможность отразить бифуркацию (от лат. *bifurcus* – раздвоение, *биолог*) – *вилкообразное* раздвоение страт по направлениям, рассматриваемым относительно независимо, но которые во взаимодействии представляют систему (ЭВМ). Таким образом, удастся дополнительно ввести программно-алгоритмические страты, имеющие ведущее прикладное значение. Результаты «тьюрингова программирования» свидетельствуют о большой популярности данной программно-алгоритмической модели решения вычислительных и функциональных задач [4].

Следующим базовым элементом стратификации можно выбрать структурно-функциональное иерархическое описание ЭВМ [5], которое в значительной своей части будем далее использовать. Между МТ и выбранными нижними стратами разрабатываемого общего стратифицированного описания вводится вербальное описание принципов построения ЭВМ фон Неймана. Страта позволяет при определенных условиях однозначно провести синтез формальной схемы ЭВМ.

## **2. Вербальная модель фон Неймана**

Страта отражает результат обобщения системных исследований вариантов построения ЭВМ, известных на середину прошлого века. Прежде всего, подразумевается физическое отделение процессорного модуля от устройств хранения программ и данных (в общей памяти). Таким образом, реализацию принципов Неймана обеспечивает модульная структура ЭВМ со связями через магистральную (системную) шину.

С современной точки зрения формированию принципов построения ЭВМ соответствуют следующие процедуры:

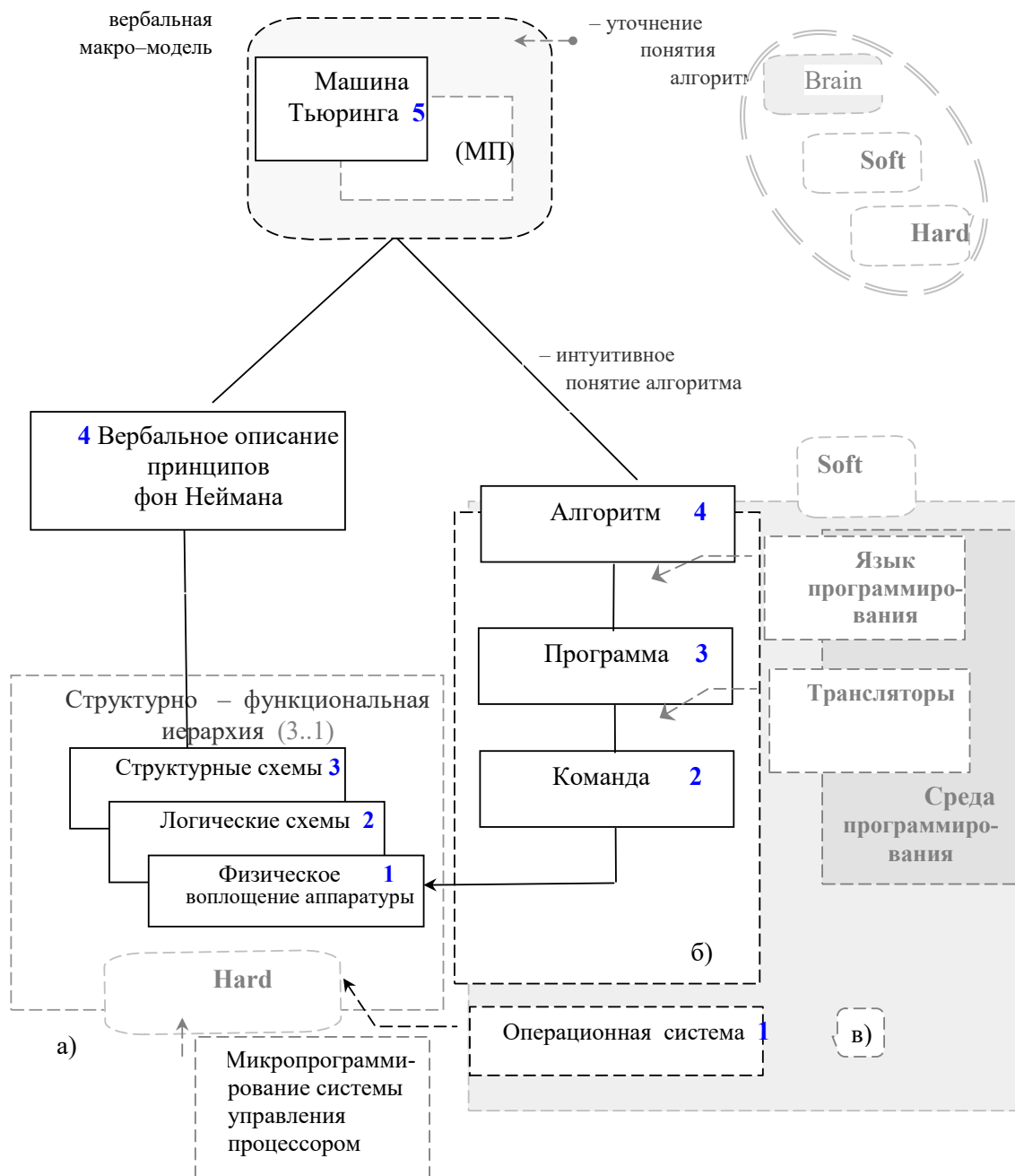
- общий анализ перспективных разработок ЭВМ;
- выявление модульной структуры (декомпозиция), детальный анализ модулей и особенностей связи через шины;
- формулировка концептуальной модели с определением структурно-функциональной направленности каждого принципа;
- синтез (сборка) схемы с контролем полноты формулировки принципов.

### **3. Единая многоуровневая модель ЭВМ**

Нижние страты отражают известный подход по формированию многоуровневой «структурно-функциональной иерархии» [5]. ЭВМ принято рассматривать на следующих уровнях: 1) электрические схемы, 2) логические схемы, 3) операционные схемы, 4) структурные схемы ЭВМ, 5) программный уровень.

Программный уровень для придания ему большей наглядности в формируемой нами модели разделен на части – прикладные компьютерные программы и системное программное обеспечение – и вынесен из приведенной цепочки. Функциональное взаимодействие страт компонентов программного уровня отражено на общей схеме стратификации (рис.1) направленными пунктирными стрелками.

Создание единой многоуровневой модели (ЕМ) ЭВМ в предлагаемой форме диктуется необходимостью видеть все проблематичные узлы семантического разрыва. ЕМ базируется на выделении важной линии стратификации прикладной компьютерной программы – от «алгоритма 4» и до «команды 2».



**Рис. 1.** Результаты стратификации для системного анализа семантического разрыва:  
 а) базовые модели; б) страты «программно-алгоритмического» уровня;  
 в) компоненты системного программного обеспечения

Функциональное воздействие команд машинного кода отражено направленной стрелкой

ЕМ сохраняет важное преимущество функциональности 5-уровневой [4] модели и явно отражается введением пунктирных линий со стрелками, исходящих от средств реализации переходов между стратами 4–3 (язык

программирования высокого уровня заданной среды программирования) и между стратами 3–2 (трансляторы), а также управляющей ресурсами связью между чрезвычайно общими стратами «Hard» – «Soft» (посредством операционной системы).

**Языковые средства.** На страте 4 разработанный алгоритм описывается человеком с помощью легко воспринимаемого графического языка. Выразительность блок-схем повышается введением комментария. Отдельные фрагменты (в соответствии с базовыми алгоритмическими структурами типа следование, ветвление, циклы) можно увидеть далее в листингах структурированной программы.

На страте 3 программа описывается с помощью абстрактных понятий организации вычислительного процесса в соответствии с языком программирования, присущим выбранной среде программирования (смотри связь на указанном уровне иерархии со средой, рис. 1 справа).

Страта 2. Языковые конструкции программы переводятся в результате трансляции в машинный код (смотри связь, рис. 1 справа). Путем формирования машинных команд в режиме компиляции готовятся исполняемые файлы.

Описание системы на каждой страте производится с помощью собственных «языковых средств». Особенности языка описания определяются не только составом конечных «пользователей» языка, но и их целями. Непосредственно организует процесс и предоставляет вычислительные ресурсы операционная система.

Таким образом, в единой модели ЭВМ задействованы три составные части – аппаратное, программное и алгоритмическое обеспечение.

#### **4. Семантический разрыв в описании и функционировании ЭВМ**

Семантическая связь всех элементов и решений дает сбалансированный проект. В ближайшей перспективе заметно возрастает значение проблемы преодоления разрыва между аппаратными средствами и методами программирования. Машины Тьюринга и Неймана – формальные модели, тем не менее, и для них с общих позиций можно обсуждать проблемы семантического разрыва. Так для МТ инструкции (программа) управляющего конечного автомата записаны в машинную таблицу. Поиск инструкций лучше осуществлять в ассоциативной памяти, которая не предусмотрена в

концепции Неймана. Налицо семантический разрыв между указанными объектами анализа. Далее, архитектура фон Неймана плохо приспособлена к программам на языках высокого уровня.

Семантический разрыв, вызванный основными конструкциями языков программирования высокого уровня, комментируется приведенными в таблице 1 данными.

Таблица 1

Объекты манипулирования	Архитектура ЭВМ IBM 370	Альтернатива ответственности	Дополнительные сведения
<b>Основные конструкции:</b> <i>массивы</i> , структуры (записи), строки фиксированной и переменной длины, процедуры, блочная структура	– редко имеет близкое «родство» с объектами и операциями, описываемыми в языках программирования		
<b>Массивы</b> – наиболее часто используемый тип организации данных в большинстве языков: <b>n-мерные</b> массивы, обращение к отдельным элементам массива с индексами <i>i, j</i> , операции над чистыми массивами, обращение – к отдельным <b>подмассивам</b> , внутри массива, защита <i>от</i> выхода <b>за границы</b> соответствующего массива, в отдельных языках – соединение массивов	не предусмотрены средства, соответствующие описанным принципам организации массивов, <b>за исключением</b> принципа использования индексных регистров. Набор команд далек по возможностям от задач, возникающих при работе с массивами	реализация широко используемого принципа организации данных в виде массивов возлагается на <b>компилятор</b>	<b>45%</b> всех арифметических операторов имеют дело по крайней мере с одним массивом или элементом массива
<b>Структуры (записи)</b>	отсутствует средство, адекватное структурам и операциям над ними	компилятор	

Продолжение табл. 1

Объекты манипулирования	Архитектура ЭВМ IBM 370	Альтернатива ответственности	Дополнительные сведения
<b>Строки</b> – обработка строк: слияние,	возможностей в архитектуре не предусмотрено	компилятор	24% всех операторов

выделение заданной части (подстроки) из исходной строки, поиск строки по заданной подстроке, определение текущей длины строки, проверка присутствия: элементов одной строки в другой строке	рено		ров присваивания предназначены для символической информации
<b>Процедуры</b> – требуется сохранение состояния текущей процедуры, динамическое назначение памяти для локальных переменных вызванной процедуры, передача параметров и инициализация выполнения вызванной процедуры	отсутствуют возможности, соответствующие этим принципам организации программ		16% выполняемых операторов языка – это обращения к подпрограммам-процедурам
<b>Блочная структура:</b> предполагает правила определения границ областей действия идентификаторов, присваиваемых объектам, определяющие возможность адресации необъявленных переменных ссылок во внутренних блоках	нет ничего подобного этому принципу адресации	реализация этого принципа должна осуществляться генерированием компилятором соответствующего машинного кода	

### Библиографический список

1. Буза М. К. Архитектура компьютеров. Минск: Новое знание, 2006. 559 с.
2. Черняк Ю. И. Системный анализ в управлении экономикой. М.: Экономика, 1975. 191 с.
3. М. Месарович, Д. Мако, И. Такахара. Теория иерархических многоуровневых систем. М.: Мир, 1973. 334 с.
4. <http://blog.wolfram.com/>
5. Майоров С. А., Новиков Г. И. Структура электронных вычислительных машин Л.: Машиностроение, Ленингр. отделение, 1979.

**В. А. Горбачев**, кандидат экономических наук, доцент,

**А. Ф. Потокин**, кандидат биологических наук, доцент

**СОЗДАНИЕ БАЗЫ ДАННЫХ ЭЛЕКТРОННОГО КАТАЛОГА  
КОЛЛЕКЦИИ «ГЕРБАРИЯ им. И. П. БОРОДИНА» СПБГЛТА.**

## АКРОНИМ (ИНДЕКС): КФТА

**Общие задачи проекта.** Согласно данным международной конференции («Preservation of botanical collections»), проводившейся в декабре 1993 г. в Санкт-Петербурге (под эгидой Юнеско), гербарной коллекции кафедры ботаники и дендрологии СПбГЛТА, насчитывающей порядка 170000 образцов сосудистых растений, присвоен ранг «Национального гербария».

Основной целью создания электронного каталога гербария государственной лесотехнической академии является получение возможности широкого доступа к информации об уникальных коллекциях гербария академии. Достижение этой цели на уровне Интернет а позволит расширить контакты ученых академии в мировом масштабе. Достижение этой цели связано с выполнением следующих этапов:

1. Создание базы данных каталога гербария ГЛТА.
2. Разработка автоматизировано поисковой системы для базы данных каталога.
3. Создание Web-страниц доступа к данным для поисковой системы.
4. Разработка сайта каталога с включением в него разработанной поисковой системы.

Первый этап включает: проектирование структуры базы данных; разработку приложения, обеспечивающего ввод сведений о каталоге в базу данных; заполнение базы данных сведениями каталога гербария.

Поисковая система должна давать возможность выбора списка интересующих данных каталога, составления из них критерия отбора и вывод результатов поиска в удобном формате. Поисковая система реализуется в виде приложения к базе данных и предназначена для «внутреннего» пользования, то есть для использования внутри компьютерной сети академии.

Web-страницы доступа к данным реализуют ту же поисковую систему, но на уровне форм, которые могут быть размещены в Интернет. И, наконец, для успешного завершения проекта необходимо было разработать сайт – «Гербарий ГЛТА», в который включаются Web-страницы поисковой системы и система обмена сообщениями типа «форум». Последняя



позволит вести активный диалог с учеными академии, организовывать контакты и встречи.

Структура базы данных разрабатывается на основе модели данных реляционного типа, то есть в виде совокупности взаимосвязанных таблиц. Каждая таблица хранит сведения о какой-либо составляющей каталога. Все таблицы базы можно разбить на группы по функциональному признаку:

- хранение систематики коллекции – семейства, рода, виды;
- хранение карточек экземпляров коллекции;
- справочники – географических названий, коллекторов и определителей, геоботанических и флористических районов и т.д.

Важную роль в структуре базы играют выбор принципов кодирования в систематике, карточках и справочниках, а так же установление связей между таблицами. Это необходимо для того, чтобы, с одной стороны, спроектировать удобное для заполнения базы данных приложение, с другой, строить запросы, позволяющие получать ответы на любые вопросы по поиску данных. Другим важным обстоятельством создания базы данных является выбор системы управления базой данных (СУБД). Для решения поставленной задачи нет необходимости использовать мощные СУБД типа ORACLE . Достаточно, чтобы СУБД обладала возможностями поддержки реляционных моделей данных в компьютерной сети, возможностью создания развитого интерфейса и создания гибких запросов к базе данных. Всеми этими свойствами обладает СУБД Access. Кроме того, это весьма распространенный и развивающийся программный продукт, снабженный большим количеством методической литературы.

Проектирование приложения требует наличия программных средств (оболочки), которая позволяла бы создавать удобные для пользователя средства работы с базой данных. Отмеченная СУБД Access полностью отвечает указанным требованиям. После создания и отладки приложения по вводу данных в базу, необходимо ввести сведения о гербарии в базу. Первоначальное заполнение базы трудоемкая, длительная и ответственная работа, которая выполняется вручную.

При разработке поисковой системы необходимо учитывать следующие требования:

- поиск экземпляров коллекции по месту сбора, коллекторам, определителям и т.д.;
- создание комбинаций условий поиска;
- представление результатов в доступной форме.

Приложения ввода данных в базу и поиск информации об экземплярах гербария, реализованные в среде Access, позволяют вести работу с каталогом гербария в рамках локальной сети. Это решение лишь части задачи. Необходимо обеспечить доступ к каталогу широкому кругу пользователей во всем мире. Этого можно достичь, создав поисковые страницы доступа к базе в среде Интернет. Access предоставляет инструментарий создания страниц доступа к данным. С его помощью на языке HTML создаются формы поиска данных аналогичные локальной поисковой системе, которые можно представить в Интернете. Для окончательного решения задачи необходимо создать сайт гербария и представить на нем странички поиска. Использование описанной информационной схемы работы с каталогом гербария выводит востребованность труда составителей гербария на мировой уровень.

### **Проектирование структуры базы данных электронного каталога гербария ГЛТА**

Структура базы данных электронного каталога гербария ГЛТА формируется на основании следующей информации:

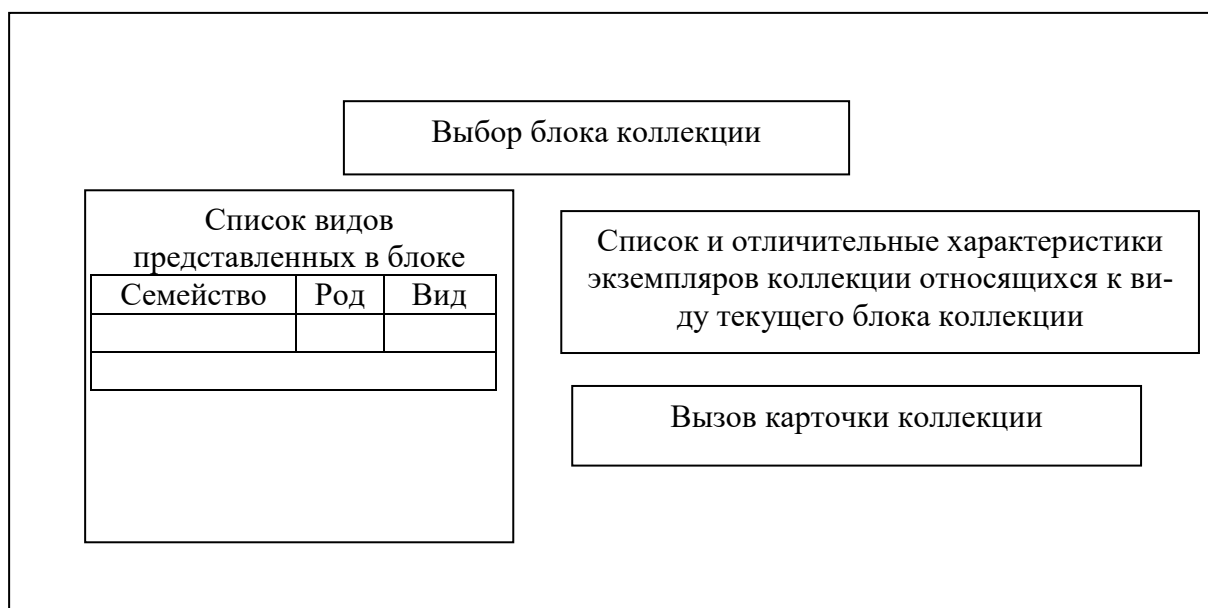
- анализа информации о существующих гербариях ГЛТА;
- анализа возможностей реализации запросов в поисковой системе;
- анализа эффективности работы приложений по вводу и поиску данных в базе.

Структура базы данных описывается как совокупность взаимосвязанных таблиц – реляционной моделью данных.

Информационной основой базы является таблица, отражающая содержимое карточки (этикетки) экземпляра коллекции и имеющая следующую структуру.

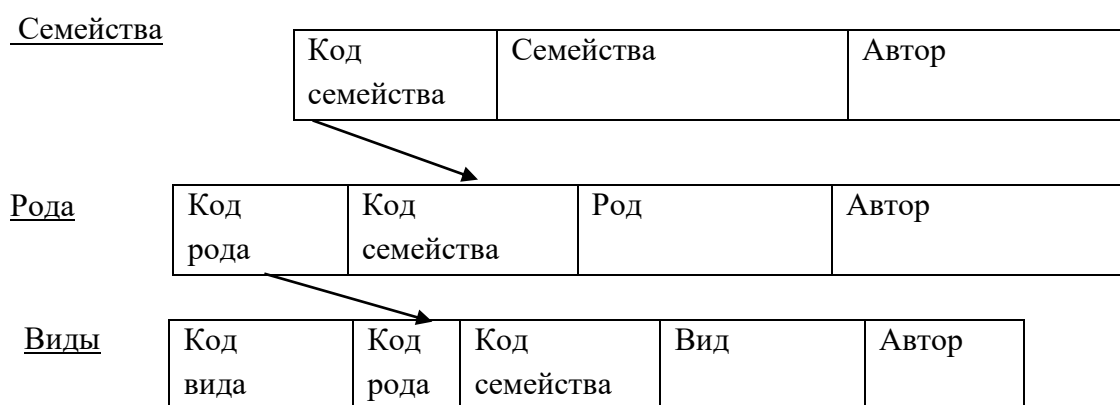
Управляющие поля карточки указывают на принадлежность карточки блоку и виду. Кроме информационной составляющей, они позволяют группировать содержимое коллекции в разрезе принадлежности класси-

кационными признакам. Эти данные позволяют реализовать следующую схему отображения информации.



**Рис. 1.** Схема отображения информации при вводе данных о коллекции

Несмотря на то, что в карточке экземпляра коллекции заносится только код вида, в экранных формах отображаются и род и семейство. Такую возможность обеспечивает следующая структура хранения информации систематики коллекции.



Данная структура играет роль справочника и позволяет хранить в карточке только код вида, при необходимости вывода на экран (печать) семейство и род вида. Поскольку экземпляров гербария, относящихся к

конкретному виду блока коллекции, может быть несколько, то в карточку введено поле с порядковым номером карточки для данного вида.

Все данные карточки представленные кодами связаны с таблицами справочниками. Коды обеспечивают связь между соответствующими таблицами. Так справочник коллекторов, имеющий следующий вид:

**Имя столбца таблицы**

— код коллектора;

— коллектор;

и связан с таблицей карточки по коду коллектора. Причем эта связь «один ко многим» со стороны справочника коллекторов. Это означает, что в справочнике каждый коллектор имеет уникальный код, т.е. не повторяется, а в таблице карточек может повторяться столько раз, сколько экземпляров гербария собрано данным коллектором. Структура таблиц справочников проста и приводится в приложении. Большинство справочников не связаны между собой. Только справочники континентов, стран, областей и населенных пунктов имеют связь между собой, т.к. имеют иерархическую подчиненность – континент → страна → область → населенный пункт. Установление связи между этими справочниками необходимо для того, чтобы выбирать страну, расположенную на континенте, область в стране и т.д. Создание справочников позволяет достичь двух целей: с одной стороны, минимизировать объем информации хранящийся в карточках. Действительно, главную часть объема данных базы составляют данные карточек и, если хранить в карточке название, например, флористического района, а не его код, то объем хранимых данных будет больше в 5–10 раз.

В общем, схема базы данных имеет следующий вид (см. рис 2).

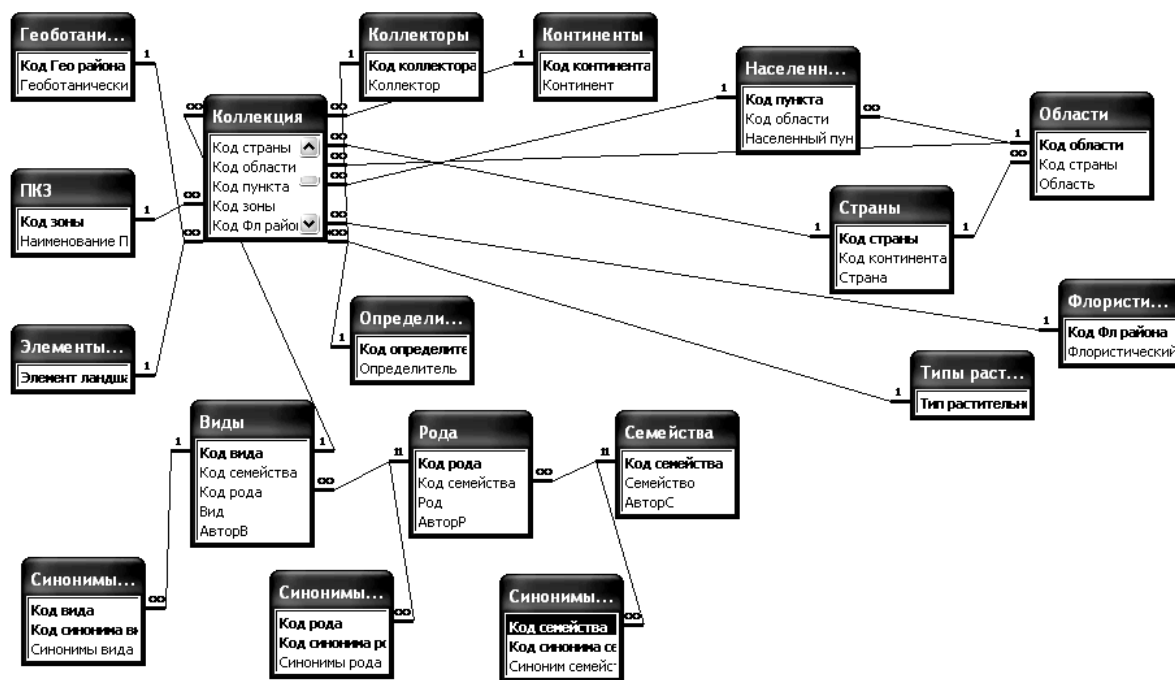


Рис.2. Схема модели базы данных электронного каталога гербария ГЛТА

Кроме описанных таблиц таблицы семейств, родов и видов имеют связанные с ними таблицы синонимов. Их наличие позволит выполнять запросы поиска альтернативных названий растений.

Работа выполнена при поддержке гранта Минобрнауки Российской Федерации (РНП.2.2.3.1.8654).

### Библиографический список

1. Лунева Н. Н., Лебедева Е. Г., Чухина И. Г. Информационно-поисковая система «Гербарий» ВИР. 1997. Павловск. 46 с.
2. Крестовская Т. В., Потокин А. Ф., Титов Ю. В. Гербарий им. И. П. Бородина Санкт-Петербургской лесотехнической академии // Известия Санкт-Петербургской лесотехнической академии. 1994. Выпуск 2 (160). С. 191–200.
3. Потокин А. Ф., Егоров А. А. Гербарий имени И. П. Бородина Санкт-Петербургской лесотехнической академии // Гербарный пресс: информационный бюллетень. №2. Ноябрь 1997.

**В. Е. Кондрашкин**, кандидат технических наук, доцент

## **РЕШЕНИЯ АЛГЕБРАИЧЕСКИХ И ТРАНСЦЕНДЕНТНЫХ УРАВНЕНИЙ ПРИКЛАДНЫМИ ПРОГРАММАМИ**

Вряд ли можно назвать математическую операцию, которую приходится применять чаще, чем вычисление значений функций по заданной формуле. В самых разнообразных задачах науки и техники встречаются математические задачи, точное решение которых невозможно получить классическими методами или оно получается настолько сложным, что становится неприемлемым для практической реализации. Разработанные вычислительной математикой численные методы носят в основном приближенный характер, получающие результат с приемлемой точностью.

Определение корней алгебраических и трансцендентных уравнений встречается в теории управления запасами, расчете налога поступления государству, исследования функциональных зависимостей между экономическими факторами, обработке статистических данных, практике расчетов линейных систем автоматического регулирования и других технических задачах.

Общая форма задания таких уравнений  $f(x) = 0$ . Алгебраические уравнения  $n$ -й степени:  $a_0x^n + a_1x^{n-1} + a_2x^{n-2} + \dots + a_n = 0$  имеют  $n$  корней, коэффициенты  $a_i$  ( $i=0,1,2,\dots,n$ ) могут быть действительными и комплексными числами. Трансцендентные уравнения включают степенные алгебраические, тригонометрические и экспоненциальные функции от некоторого аргумента  $x$ , например,  $\arctg(x) - x + 1 = 0$ . Такие уравнения обычно имеют бесконечное множество корней.

Программы решения уравнений чаще всего используют методы последовательных приближений, или итерационные, сводящиеся к последовательному уточнению начального значения корня. Для алгебраических и трансцендентных уравнений используются одни и те же методы. Отыскание начального приближения корней составляет проблему отделения корней. Обычно область отыскания корней определена постановкой задачи, поскольку речь идет о главных значениях аргументов. Для отделения дей-

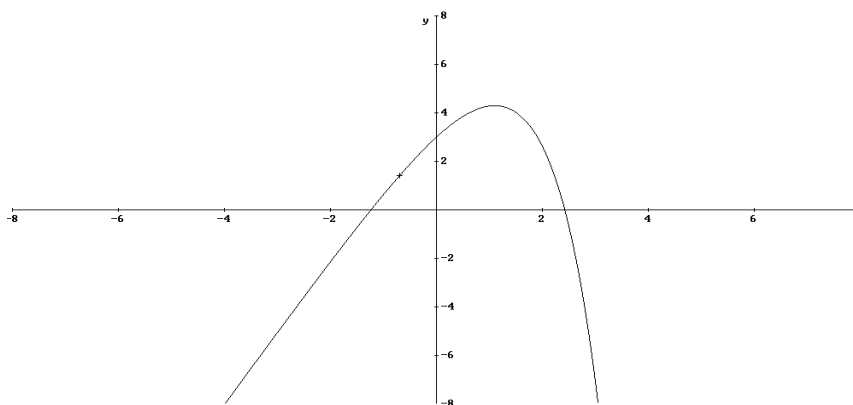
ствительных корней часто достаточно построить графики изменения функции  $f(x)$  в выбранных границах изменения аргумента.

В настоящей работе рассматривается возможность нахождения корней уравнений на конкретных примерах в системе символьной математики Derive, математическом пакете Mathcad и программе обработки электронных таблиц Excel.

Для иллюстрации и анализа численного решения уравнения рассматриваемыми программами возьмем формулу:  $Y=3x - e^x + 4$ . Среди многообразия математических пакетов и систем автоматизации расчетов Derive отличается тем, что в нем удачно сочетаются возможности проведения численных и символьных вычислений с простотой и сравнительно не высокими требованиями к техническим характеристикам используемой техники. Фактически, это единственно компактная математическая система, обеспечивающая решение задач компьютерной алгебры в диалоговом режиме с приемлемым быстродействием и хорошими графическими возможностями.

Решение уравнения  $y=3x-\exp(x)+4$  в Derive

1. *Графическое решение* иллюстрируется следующим образом:



2. *Аналитическое решение.* Из семи типовых вариантов решения уравнений выбрали команду определения численного значения корня в заданной области изоляции: NSOLVE ( $3 \cdot x - \text{EXP}(x) + 4 = 0$ ,  $x$ ,  $-2, 0$ ), при этом первый корень уравнения в левой точке пересечения с осью абсцисс  $x_1 = -1.236537334$ . Корень уравнения в области изоляции  $[2,2; 2,5]$   $x_2 = 2.421712881$ .

В практической деятельности для решения математических задач часто используется физико-математический пакет Mathcad с включенной системой искусственного интеллекта Maple, которая позволяет выполнять математические вычисления не только в числовой, но и в аналитической (символьной) форме. Особое внимание разработчики Mathcad уделили удобству работе с ним и простоте освоения. Интерфейс прост и понятен, полностью отвечает стандартам среды Windows. Все графики и математические объекты могут быть введены щелчком мыши с перемещаемых палитр. Обучение пользователя происходит в процессе работы «на ходу» при помощи многочисленных сообщений системы.

Графическая среда Mathcad позволяет записывать математические формулы в привычном виде, гибко и выразительно представлять данные графически.

В Mathcad встроено девять типов графиков. Причем деление на типы несколько условно т.к. управляя установками многочисленных параметров, можно создавать комбинации типов графиков, а также новые типы (например, двумерная гистограмма распределения является разновидностью просто X-Y графика).

Самый простой и наглядный способ получить декартов график (X-Y график) – это сформировать два вектора данных, которые будут отложены вдоль осей X и Y.

Для заданной функции  $y(x) := 3 \cdot x - e^x + 4$  зададим вектор значений переменной оператором ранжирования. В результате переменная и функция будут заданы в виде двух соразмерных векторов, по которым построен график.

На графике отложены точки, соответствующие парам элементов векторов, соединенные отрезками прямых линий. Образованная ими ломанная называется *рядом данных* или *кривой*, при этом Mathcad автоматически определяет границы графика, исходя из диапазона значений элементов векторов.

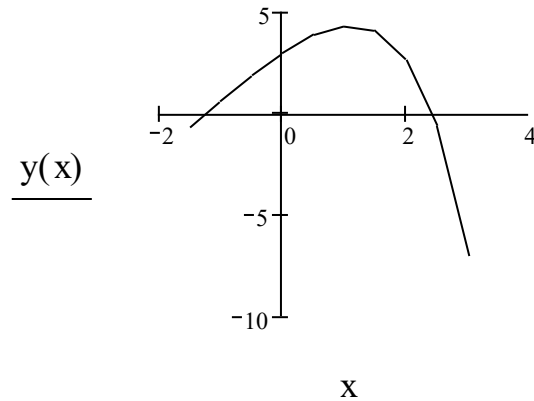


$$y(x) := 3 \cdot x - e^x + 4$$

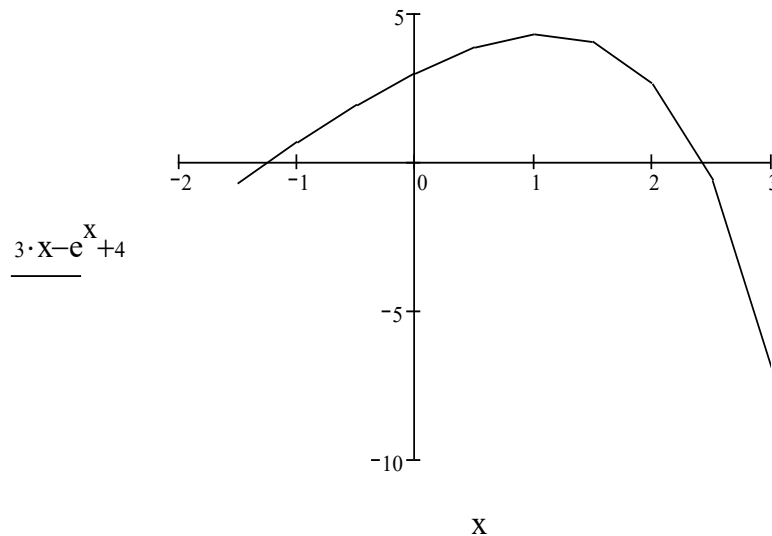
X := -1.5, -1 .. 3

x =      y(x) =

-1.5	-0.723
-1	0.632
-0.5	1.893
0	3
0.5	3.851
1	4.282
1.5	4.018
2	2.611
2.5	-0.682
3	-7.086



Нарисовать график любой скалярной функции  $f(x)$  можно более простым способом, называемым *быстрым построением графика*, который заключается во введении функции в один из местозаполнителей (в нашем примере, у оси Y), а имя аргумента – в местозаполнитель у другой оси.



Как известно, точки пересечения графика с осью абсцисс – точки, в которых функция  $y(x) = 0$  или, по крайней мере, не превышает заданной точности вычисления – определяют вещественные корни уравнения.

Следующим шагом решения является уточнение графически определенных корней численными методами с заданной погрешностью (в Mathcad ее роль играет специальная константа TOL).

Для решения уравнения с одним неизвестным, кроме вычислительного блока *Given/Find*, предусмотрена встроенная функция *root*, которая может включать, в зависимости от типа задачи, либо два, либо четыре аргумента.

$$\begin{array}{lll}
 y(x) := 3 \cdot x - e^x + 4 & x := -1.5 & \text{TOL} := 10^{-4} \\
 & \text{root}(y(x), x) = -1.2365 & \\
 & x := 2 & \\
 & \text{root}(y(x), x) = 2.4217 &
 \end{array}$$

Точность вычислений по умолчанию принимается  $\text{TOL}=10^{-3}$ .

$$\begin{array}{ll}
 x1 := \text{root}(y(x), x, -1.5, -0.5) & x1 = -1.237 \\
 x2 := \text{root}(y(x), x, 2, 3) & x2 = 2.422
 \end{array}$$

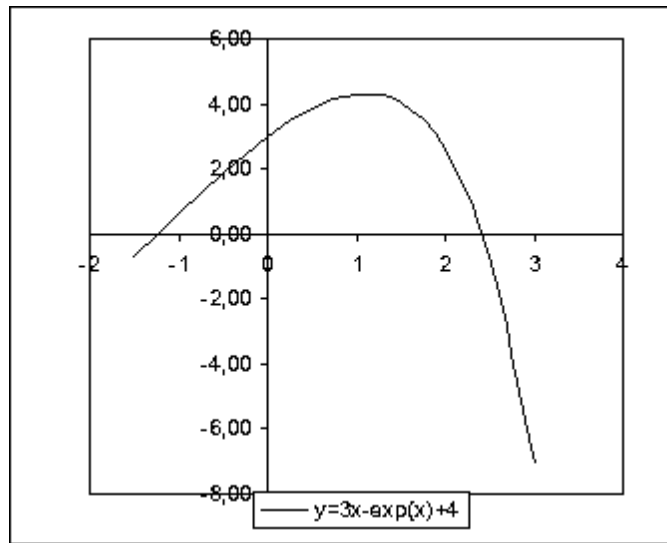
Алгоритм функции *root* реализует численный метод – секущих (хорд). Во встроенной функции *Find* реализовано несколько градиентных численных алгоритмов, один из которых может выбрать либо программа Mathcad, либо пользователь. Принцип градиентных методов состоит в последовательном приближении к истинному решению уравнения, которые вычисляются с помощью производной от  $f(x)$ . Наиболее простая форма алгоритма – метод Ньютона (касательных).

В аномальных случаях, например, попытка применить метод секущих в области локального максимума или минимума функции. В этом случае секущая может иметь направление близкое к горизонтальному – решения нет. Для решения таких уравнений применяют встроенную функцию *Minerr*. Она применяется аналогично функции *Find*, в частности, имеет тот же самый набор параметров, реализует те же самые алгоритмы, но критерием завершения работы является схождение итерации к минимальным значениям исследуемой функции.

В настоящее время Microsoft Excel представляет собой достаточно мощное средство разработки информационных систем, которое включает как электронные таблицы (со средствами финансового и статистического анализа, набором стандартных математических функций, рядом дополнительных функций, встречающихся только в библиотеках дорогостоящих

инженерных подпрограммах), так и средства визуального программирования. Электронные таблицы позволяют производить обработку чисел и текста, задавать формулы и функции для автоматического выполнения, представлять данные в виде графиков и диаграмм. Для построения графика функции, с целью нахождения корней, вычислим таблицу табуляции в области  $[-1,5; 3]$  с шагом 0,5.

№	x	$y=3x-\exp(x)+4$
1	-1,5	-0,72
2	-1	0,63
3	-0,5	1,89
4	0	3,00
5	0,5	3,85
6	1	4,28
7	1,5	4,02
8	2	2,61
9	2,5	-0,68
10	3	-7,09



По графику определяем области изоляции отрицательного  $[-1,5; -0,9]$  и положительного  $[2,3; 2,6]$  корней.

Уточняем корни уравнения методом хорд с точностью  $\varepsilon=0,0001$  по условию:

$$d = \text{abs}(x(i) - x(i+1)) \leq \varepsilon \text{ или } y(x)=0.$$

Отрицательный корень:

№ п/п	$x = x - f(x) * (x-a) / (f(x) - f(a))$	$y = 3x - \exp(x) + 4$	$d = \text{abs}(x(i) - x(i+1))$
1	-0,9	0,89343	
2	-1,23160	0,01336	0,33160
3	-1,23647	0,00017	0,00487
4	-1,23654	0,00000	0,00006

Принимаем из таблицы  $x_1 = -1,2365$ .

Положительный корень:

№	$x = x - f(x) * (b - x) / (f(b) - f(x))$	$y = 3x - \exp(x) + 4$	$d = \text{abs}(x(i) - x(i+1))$
1	2,3	0,92582	
2	2,40726	0,11832	0,10726
3	2,42005	0,01370	0,01280
4	2,42152	0,00157	0,00147
5	2,42169	0,00018	0,00017
6	2,42171	0,00002	0,00002

Принимаем из таблицы  $x_2 = 2.4217$

В заключение отметим, что результат решения заданного уравнения рассматриваемыми программами с точностью до четырех знаков после запятой одинаков. Самым простым безусловно является Derive, но простота ограничивает возможности. Наиболее универсальным, емким и красивым в решении уравнений из рассматриваемых пакетов можно назвать Mathcad.

#### Библиографический список

1. *Кирьянов Д.* Mathcad 12. СПб.: БХВ-Петербург, 2005. 557 с.
2. *Ланджер М.* Microsoft office EXCEL 2003 для windows, 2005, NT Press. 365 с.
3. *Половко А. М.* Математическая система Derive для студентов. СПб.: БХВ-Петербург, 2005. 339 с.

**Т. В. Коваленко**, кандидат технических наук

## **ПРИМЕНЕНИЕ ИНФОРМАЦИОННЫХ СИСТЕМ ДЛЯ РАЦИОНАЛЬНОЙ ОРГАНИЗАЦИИ ЛЕСНЫХ ГРУЗОПОТОКОВ**

Существующая система лесосечно-транспортного процесса в лесозаготовительных предприятиях предлагает годовое, квартальное и месячное его планирование.

Особо актуальной в связи со сложившейся системой организации лесозаготовок является месячное планирование лесозаготовок и транспорта леса, которое носит массовый характер. Оно осуществляется на уровне планово-производственных отделов предприятия и предполагает определение не только объемов, но и местоположение лесозаготовок, объемов складирования, хранения и сроков вывозки заготовленного леса, что, в конечном счете, в значительной мере определяет эффективность всего лесозаготовительного процесса.

Существующая практика планирования последовательности освоения годового лесосечевого фонда содержит множество неформализованных субъективных этапов, в основном основанных на интуиции и опыте предыдущих лет.

Наиболее плодотворным путем решения задач такого рода является разработка экономико-математических моделей и их реализация на ЭВМ с принятием обоснованных плановых и проектных решений.

Лесотранспортные работы в лесозаготовительном производстве имеют определяющее значение. От правильного и своевременного выполнения лесотранспортных работ зависят все последующие операции и конечные результаты работы ЛЗП в целом. По содержанию и условиям выполнения лесной транспорт значительно отличается от других фаз технологического процесса лесозаготовок. Их особенностью является существенная зависимость от природных условий: погоды и климата, грунтово-гидравлических условий и таксационных характеристик лесосек. Наибольшее влияние на лесотранспортный процесс оказывают погодноклиматические факторы, что в первую очередь и определяет сезонность

лесозаготовок. Значительная изменчивость этих характеристик, как по времени (сезонные колебания), так и в пространстве (каждая лесосека имеет свои условия), делает задачу проектирования лесосечно-транспортных процессов чрезвычайно сложной.

На современном этапе развития лесозаготовительного производства можно выделить три основных слагаемых оптимального решения задачи организации механизированных работ на группе объектов производственной программы ЛЗП:

- решение вопроса о рациональном пространственно-временном освоении годового лесосечного фонда;
- выбор наиболее эффективных способов выполнения работ на объектах программы планируемого периода (выбор принципиальной схемы разработки лесосек, рациональной конструкции технологических путей).

Максимальный эффект дает лишь совместное решение всех указанных задач.

Этого можно достичь путем создания интегрированных систем автоматизированного проектирования и управления лесопользованием.

Такие методы и автоматизированные системы разрабатываются и внедряются, однако большинство из них рассчитаны на решение стратегических задач в рамках региона, области или холдинга и поэтому не всегда применимы к каждому конкретному лесозаготовительному предприятию.

Содержательное описание задачи сводится к следующему. Имеется годовой лесосечный фонд лесозаготовительного предприятия, сведения о котором включают общее число лесосек, таксационное описание каждой из них, их географическое местоположение и грунтово-гидрологические условия.

Все лесосеки должны быть освоены в течение планируемого года, причем любая из них может быть разработана в любое время в течение года, но расходы на лесозаготовки, лесовосстановление и лесотранспорт будут при этом различны.

Заготовленная на лесосеках древесина может быть отправлен сразу же (не позднее месяца со дня заготовки) на нижний склад, либо складирован здесь же в штабелях с последующей вывозкой его на нижний склад или потребителю в наиболее рациональные календарные сроки. Эффектив-

ность лесосечно-транспортных работ так же зависит от сезона года, поскольку расходы на лесосечные работы, дорожное строительство, а так же транспортная составляющая себестоимости вывозки значительно отличаются по сезонам года.

Считается так же, что месячные поставки леса потребителю определены и известны заранее. Месячные объемы заготовки и складирования древесины на лесосеках, промежуточном и нижнем складе не определены; известен лишь их верхний предел, установленный наличием складских площадей, материально-технических и трудовых ресурсов.

Математическая модель поставленной задачи имеет следующую целевую функцию:

$$\begin{aligned}
 F = & \sum_{i=1}^m \sum_{k=1}^{12m} C_{ik} X_{ik} + \sum_{i=1}^m \sum_{l=1}^{12p} C_{il} X_{il} + \sum_{i=1}^m \sum_{r=1}^{12d} C_{ir} X_{ir} + \sum_{k=1}^{12m} \sum_{l=1}^{12p} C_{kl} X_{kl} + \\
 & + \sum_{k=1}^{12m} \sum_{r=1}^{12d} C_{kr} X_{kr} + \sum_{k=1}^{12m} \sum_{j=1}^{12n} C_{kj} X_{kj} + \sum_{l=1}^{12p} \sum_{r=1}^{12d} C_{lr} X_{lr} + \sum_{l=1}^{12p} \sum_{j=1}^{12n} C_{lj} X_{lj} + \\
 & + \sum_{r=1}^{12d} \sum_{j=1}^{12n} C_{rj} X_{rj} \rightarrow \min.
 \end{aligned} \quad (1)$$

Задача содержит следующие ограничения.

Общие запасы годового лесосечевого фонда должны быть равны или превосходить заключенные предприятием контракты на поставку древесины

$$\sum_{i=1}^m a_i + \sum_{k=1}^{12m} q_k + \sum_{l=1}^{12p} S_l + \sum_{r=1}^{12d} N_r > \sum_{j=1}^{12n} b_j. \quad (2)$$

Сумма поставок  $j$ -му потребителю из всех лесосек, отведенных в рубку должна быть равна спросу этого потребителя

$$\sum_{i=1}^N x_{ij} + \sum_{k=1}^{12m} x_{kj} + \sum_{l=1}^{12p} x_{lj} + \sum_{r=1}^{12d} x_{rj} = b_j \quad (j = \overline{1, 12n}). \quad (3)$$

Сумма поставок леса с каждой  $i$ -й лесосеки на верхний сезонный, нижний склад и во двор потребителя не должна быть более ликвидного запаса в этой лесосеке

$$\sum_{k=1}^{12m} x_{ik} + \sum_{l=1}^{12p} x_{il} + \sum_{r=1}^{12d} x_{ir} + \sum_{j=1}^{12n} x_{ij} \leq a_i \quad (i = \overline{1, m}) \quad (4)$$

Суммарный объем поставок леса на верхние склады должен быть равен мощности каждого из них, а поставки из этих складов на сезонный склад, нижний склад и во двор потребителю не должны превосходить установленного верхнего предела их мощности

$$\sum_{i=1}^m x_{ik} = X_k = \sum_{l=1}^{12p} x_{kl} + \sum_{r=1}^{12d} x_{kr} + \sum_{j=1}^{12n} x_{kj} \leq q_k \quad (k = \overline{1, 12m}) \quad (5)$$

Суммарный объем поставок леса на сезонные склады с лесосек и верхних складов должен быть равен мощности каждого из них, а поставки с сезонных складов на нижний склад и потребителем не должны превышать установленного верхнего предела их мощности

$$\sum_{i=1}^m x_{il} + \sum_{k=1}^{12m} x_{kl} = X_l = \sum_{r=1}^{12d} x_{lr} + \sum_{j=1}^{12n} x_{lj} \leq S_l \quad (l = \overline{1, 12p}) \quad (6)$$

Суммарный объем поставок леса на нижние склады с лесосеки, верхних и сезонных складов должен быть равен мощности каждого из них, а поставки с нижних складов во двор потребителю не должны превосходить установленного верхнего предела их мощности

$$\sum_{i=1}^m x_{iq} + \sum_{k=1}^{12m} x_{kr} + \sum_{l=1}^{12p} x_{lr} = X_r = \sum_{j=1}^{12n} x_{rj} \leq N_r \quad (r = \overline{1, 12d}) \quad (7)$$

Задача содержит так же ограничение на неотрицательность всех неизвестных

$$\begin{aligned} x_{ik} &\geq 0; & x_{il} &\geq 0; & x_{ir} &\geq 0; & x_{ij} &\geq 0; \\ x_{kl} &\geq 0; & x_{kr} &\geq 0; & x_{kj} &\geq 0; & x_{lr} &\geq 0; \\ x_{lj} &\geq 0; & x_{rj} &\geq 0; & (i = \overline{1, m}; & k = \overline{1, 12m}; & l = \overline{1, 12p}; & r = \overline{1, 12d}; & j = \overline{1, 12n}) \end{aligned} \quad (8)$$

где  $x_{ik}$  – объем леса, вырубленный на  $i$ -той лесосеке и складированный на ней для последующей вывозки;  $x_{il}$ ,  $x_{ir}$ ,  $x_{ij}$  – объем леса, вырубленный на  $i$ -той ле-



сосеке с доставкой на промежуточный склад, нижний склад и во двор потребителя соответственно;  $x_{kl}$ ,  $x_{kr}$ ,  $x_{kj}$  – объем леса, вывозимый с верхнего склада соответственно на промежуточный склад, нижний склад и во двор потребителя;  $x_{lr}$ ,  $x_{lj}$  – объем леса, вывозимый с промежуточного склада на нижний склад и во двор потребителя;  $x_{rj}$  – объем леса, вывозимый с нижнего склада потребителю.

Предложенная модель, метод ее преобразования и построения сводной матрицы исходных данных обладают достаточной универсальностью и легко преобразовываются для большего количества звеньев в системе лесотранспорта. На этом основании на кафедре Сухопутного транспорта леса Лесоинженерного факультета под руководством профессора Н. А. Тюрина была создана система OZVD.

Исходными данными для системы являются сведения о годовом лесосечном фонде ЛЗП. Каждая лесосека характеризуется ликвидным объемом, площадью общей и неэксплуатационной, геометрическими параметрами, наличием подроста, расстоянием до существующей дороги, типом местности по условиям увлажнения, средним объемом хлыста и расстоянием вывозки до нижнего склада.

Материально-технические ресурсы ЛЗП характеризуются системой лесосечно-транспортных машин, месячными объемами заготовки и вывозки леса, а так же месячными потребностями нижнего склада в древесине.

Выходная информация, полученная в результате работы системы, выводится как на дисплей, так и на принтер, а так же в файл. Она содержит матрицу решения транспортной задачи, таблицу календарных графиков заготовки и вывозки леса с каждой из лесосек годового лесосечного фонда, таблицу рациональных схем и конструкций дорожных одежд на усах, количество погрузочных пунктов на лесосеке и принципиальные схемы трелевочных волоков, а так же технологическую карту разработки лесосеки с технико-экономическими показателями.

Использование системы в учебном процессе и опытное внедрение на производстве показало ее работоспособность и эффективность.

## **ОПЫТ ПРОВЕДЕНИЯ И УЧАСТИЯ В ОЛИМПИАДАХ ПО ИНФОРМАТИКЕ И ПРОГРАММИРОВАНИЮ КАФЕДРЫ ИНФОРМАТИКИ И ИНФОРМАЦИОННЫХ СИСТЕМ**

Подготовка студентов, имеющих хорошие знания в области информатики и информационных технологий – одна из задач важных Высшей школы. Именно поэтому в последние годы кафедра информатики и информационных систем ежегодно проводит олимпиады по информатике для студентов всех факультетов. Цель таких олимпиад, с одной стороны выявить способных студентов, интересующихся различными вопросами информатики, с другой – повысить общий уровень подготовки по информатике и информационным технологиям. Правда, в условиях вуза, где информатика для большинства студентов является просто общеобразовательным предметом, проведение таких олимпиад имеет свои особенности. Поступая в академию, большинство студентов не планируют активно использовать знания по информатике и информационным технологиям в своей будущей практической деятельности и, как правило, имеют весьма посредственную подготовку по информатике. Умение общаться в Интернете и использовать электронную почту не в счет, это скорее дань времени, чем итог серьезной подготовки по информатике в средней школе.

Первая олимпиада по информатике для студентов всех курсов и факультетов была организована кафедрой информатики и информационных систем в ноябре 2002 года. Олимпиада проводилась в двух номинациях: теоретической и практической, на каждую отводилось по два астрономических часа. Студент по выбору мог участвовать в обеих номинациях или в одной. Задания по теоретической информатике выполнялись в письменной форме, а для практической информатики участнику предоставлялся компьютер, где он за отведенное время должен был правильно выполнить максимальное количество заданий. Задания по теоретической информатике были подобраны так, чтобы охватить основные разделы теоретической информатики, изучаемые в курсе информатики средней и высшей школы. Были предложены задания по системам счисления, кодированию инфор-

мации, логике, алгоритмизации, основам программирования. Сложность заданий была различной: от простых, для которых достаточна подготовка в объеме средней школы или просто эрудиции, до сложных, требующих не только знаний, но и сообразительности. В каждом задании указывалось максимальное количество баллов, которое может получить участник, чем сложнее задание – тем больше баллов оно давало участнику при правильном решении задачи.

В заданиях по практической информатике нужно было грамотно использовать возможности Microsoft Office при подготовке различных документов, рефератов, организации технических и экономических расчетов; требовалось умение выполнить математические расчеты с помощью предложенного математического пакета или электронных таблиц.

Первый опыт оказался удачным, хотя участников было немного – около двадцати человек. Победителем олимпиады стала студентка первого курса лесоинженерного факультета, которая действительно имела основательную подготовку по всем разделам информатики, и как выяснилось, хотела бы учиться по специальности, связанной с информационными технологиями. При поступлении на первый курс она не решилась поступать на специальность из-за большого конкурса. Заведующий кафедрой поддержал эти пожелания и ходатайствовал перед администрацией академии о переводе на специальность «Информационные системы и технологии».

Следующая олимпиада проводилась по тем же классическим правилам, победителями становились три участника, набравшие максимальное количество баллов. Но кафедре хотелось, чтобы олимпиада привлекала большее число студентов, способствовала лучшей подготовке будущих специалистов не только увлекающихся информатикой, но и тех, кто еще не понял, что это интересная и полезная дисциплина. Хотелось, чтобы, прочитав объявление об олимпиаде, студент не пробежал мимо, а перелистал конспекты и учебники, вспомнил лекции и практические занятия и пришел оценить свои знания и навыки по информатике и информационным технологиям.

Для этого в правила олимпиады были внесены дополнения: все желающие сдать досрочно экзамен по информатике, могут принять участие в олимпиаде, выполнить обязательные задания по теоретической информатике, помеченные звездочкой, набрать при этом количество баллов не

меньше указанного, тогда в сессию им придется явиться с зачеткой для получения оценки. Оценки выставлялись только две – пять или четыре, при условии, что получен зачет по практическим занятиям и прослушаны все лекции в соответствии с учебным планом. Претенденты, которым не удалось набрать нужные баллы, без всяких последствий продолжали изучать информатику и сдавали экзамен по курсу «Информатика» на общих основаниях. Сроки проведения олимпиады для этого условия были самые подходящие: ноябрь, а экзамен по информатике сдается на втором курсе, зимой. При желании студент мог получить подробный перечень тем, которые рассматриваются в олимпиадных задачах. Количество участников олимпиад после этого удвоилось.

В таком режиме олимпиада проводилась и в этом учебном году. Участников олимпиады было более сорока. Победителями олимпиады стали студенты, обучающиеся по специальности «Информационные системы и технологии». Экзаменационную оценку по результатам олимпиады удалось получить шести студентам ФЭУ. Надо отметить, что наиболее активными участниками олимпиады оказались студенты лесохозяйственного факультета и факультета экономики и управления. Это объясняется не только активностью студентов, но и преподавателей, которые не полагались на силу вывешенных объявлений, а на лекциях и практических занятиях пытались привлечь студентов к участию в олимпиаде.

Задания постоянно дополняются и уточняются, теоретическая информатика по-прежнему принимается в письменном виде, так как модные сейчас тесты в электронной форме не позволяют правильно оценить возможности участника, его умение мыслить, строить алгоритм и рассуждать. Ведь правильно разработанная задача, пусть с незначительной ошибкой в результатах, и случайно угаданный в тесте ответ, не одно и то же. Такой подход добавляет работы и хлопот жюри, но позволяет лучше выявить способных студентов.

Опыт проведения олимпиад позволил сделать следующие выводы:

- в олимпиаде чаще принимают участие студенты младших курсов. Это является положительным фактором, так как обнаруженные дарования имеют достаточно времени, чтобы научиться применять свои знания в выбранной отрасли лесного хозяйства;

- уровень подготовки по информатике в средней школе растет, хотя и медленно;
- анализ результатов позволяет оценить, каким разделам информатики нужно уделить больше внимания при теоретической и практической подготовке студентов;
- разработанные задания могут быть полезны при проведении практических и лабораторных занятий по курсу информатики.

Кафедра имеет опыт подготовки и участия в международных олимпиадах и начала приобретать его осенью 2003 года, когда Лесотехническая академия получила приглашение принять участие в четвертьфинале командного чемпионата мира по программированию. Оно было передано на кафедру информатики и информационных систем, где после некоторых раздумий было решено попробовать создать команду и принять участие в соревнованиях.

Над чем задуматься было, это серьезнейшие соревнования, командный чемпионат мира по программированию (ICPC, International Collegiate Programming Contest), которые проводятся ежегодно под эгидой американской Ассоциации по вычислительной технике (ACM, <http://www.acm.org>), цель чемпионата – отобрать студентов, способных профессионально заниматься программированием. Об итогах чемпионата мира по программированию мы слышим ежегодно весной, когда в новостных передачах с удовольствием слышим о победах команд СПбГУ, СПбГИТМО или МГУ. Но такие команды могут создать ВУЗы, которые готовят профессиональных программистов, а своих будущих студентов отбирают еще в средних специализированных школах и лицеях.

Чемпионат мира проводится обычно в конце учебного года в апреле, отбор участников начинается в начале учебного года: в ноябре по регионам проводят четвертьфиналы, затем через месяц – полуфиналы. После чего победители отправляются готовиться к финальному бою, а побежденные продолжают работать, помня, что очередные отборочные четвертьфинальные соревнования не за горами.

При подборе участников нужно было учитывать следующие правила: команда должна состоять обязательно из трех человек, официальный язык общения на олимпиаде – английский, т. е. общение с жюри по вопросам

олимпиады и текст всех заданий предлагаются только на английском языке. Академия не готовит профессиональных программистов, поэтому среди наших студентов могут быть только любители, которых еще надо обнаружить. Команду должны объединять не только знания по алгоритмизации и программированию, но и общий язык программирования, так как команда выбирает для работы один из предлагаемых языков программирования (например, в этом сезоне JAVA, C, C++, PASCAL).

В условиях таких ограничений и психологических сложностей (обнаруженные таланты иногда отказываются от участия, услышав о ранге соревнований) была собрана команда. Первыми участниками Северного четвертьфинального подрегиона чемпионата в 2003 году стали студенты разных курсов, обучающиеся по специальности «Информационные системы и технологии» Голофеев С., Веретешкин М., Рогозина М. Первый опыт оказался неплохим, команде удалось сдать одну задачу, хотя была попытка сдать три задания из одиннадцати.

По правилам соревнований задача считается сданной, если предоставленная командой работающая программа отвечает всем требованиям тестовой системы. Для каждой задачи указывается максимальное время выполнения для одного теста и максимальный объем памяти, который может быть использован программой. Если на одном из тестов программа превысила это время или ограничение по памяти, то решение считается неверным. Если решение не зачтено, сообщается тип ошибки и номер теста, на котором ошибка произошла. Все тесты выполняются в одном и том же порядке с возрастанием сложности. Первые тесты обычно совпадают с примерами из условия задачи. Номер ошибочного теста, указываемый в результатах проверки, есть номер первого теста, не прошедшего проверку. Таким образом, отбираются программы с эффективными алгоритмами.

Первый опыт был продолжен, в прошлые годы академия представила две команды, а в этом году три. Участниками становились студенты, обучающиеся по специальности «Информационные системы и технологии», а также будущие экономисты и технологи.

Были бессменные участники, такие как Окулов А., Вагапов В. Савельев Б., Алексенко А., нашим командам удавалось сдать по заданию, что в условия таких соревнований очень неплохо. За это время участники повысили свой профессиональный уровень, изучив почти самостоятельно

язык программирования JAVA, расширили знания по алгоритмизации, да просто приобрели опыт умения работать в команде, навык участия в больших соревнованиях, что может быть полезным в дальнейшей профессиональной деятельности. Но время идет, участники заканчивают курс обучения и покидают академию, поэтому поиск новых талантов продолжается постоянно.

**М. О. Лебедев**, кандидат технических наук, доцент

## **ПРИНЦИПЫ СОЗДАНИЯ ДИНАМИЧЕСКИ НАСТРАИВАЕМЫХ ИНФОРМАЦИОННЫХ СИСТЕМ**

Достаточно часто информационные системы могут быть представлены в виде совокупности трех составляющих:

- 2) основной список;
- 2) информационный блок;
- 3) журналы.

Под основным списком будем понимать информацию, параметры которой (состав полей) не меняется (или меняется крайне редко). Каждая запись основного списка определяет основную единицу хранения в информационной системе (ИС). Например, список студентов ВУЗа, список учащихся школ, список пациентов больницы, список производимой продукции, список товаров и т.д. Основной список представляет собой высоко нормализованную (не ниже третьей нормальной формы) таблицу базы данных. Состав полей этой таблицы практически неизменяемый.

Под информационным блоком будем понимать некую совокупность параметров, описывающих каждый элемент основного списка. Состав параметров может изменяться при переходе от одного элемента основного списка к другому. Этот блок содержит актуальную информацию и не содержит историю изменения этой информации. Примеры информационных блоков:

для студентов вуза:

- 1) номер студенческого билета,
- 2) тип документа о среднем образовании (школьный аттестат или диплом профессионального училища),

- 3) номер аттестата/диплома о среднем образовании,
- 4) адрес и т.д.;

для списка пациентов больницы:

- 1) дата поступления,
- 2) диагноз при поступлении,
- 3) дата выписки,
- 4) отделение,
- 5) номер палаты и т.д.;

для списка производимой продукции:

- 1) документ(ы), на основании которого ведется выпуск продукции,
- 2) список главных лиц, отвечающих за выпуск продукции (главный конструктор, главный технолог и т.д.) и т.д.

Под журналом будем понимать некую совокупность параметров, каждое изменение которой регистрируется как отдельная запись. Совокупность параметров связана с элементами основного списка. Состав полей журнала определен. Этот блок содержит историю (движение) параметров, описывающих каждый элемент основного списка. Например:

для студентов вуза:

- 1) результаты экзаменационных сессий,
- 2) движение студента (с курса на курс, с одной формы обучения на другую и т.д.);

для списка пациентов больницы:

- 1) результаты анализов,
- 2) результаты осмотра врачей;

для списка производимой продукции:

- 1) объем выпуска продукции по месяцам (отчетным периодам),
- 2) объемы поставок продукции,
- 3) поступившие рекламации,
- 4) реакция на рекламации.

В блоке «Журналы» в ходе эксплуатации ИС может состав журналов может изменяться (добавляться новые, удаляться ранее созданные журналы).

Под динамически настраиваемой ИС будем понимать программный комплекс, включающий собственно данные и программные компоненты для работы с этими данными, позволяющие:



1) изменять состав параметров информационного блока – добавлять новые параметры, удалять более не нужные, изменять способ ввода значений для этих параметров (ввод вручную, выбор одного или нескольких значений из списка);

2) изменять состав журналов;

3) создавать обработчики событий при работе с составляющими ИС.

При этом ранее созданные программные компоненты не должны перепрограммироваться, не должен меняться интерфейс приложений, работающих с этой базой данных. Допускается создание новых программных компонент, включение которых в программный комплекс не должно вызывать конфликта с ранее созданными программными компонентами.

Принципы создания ИС, отвечающей этим требованиям, рассмотрим на примере создания программного комплекса (ПК) для больницы. Требуется разработать ПК, регистрирующий поступление и выписку больных, результаты взятых анализов, осмотр врачей, назначенное лечение. При этом часть больных может находиться в стационаре круглосуточно, а другая часть – только днем.

Основной список в данном случае может быть представлен следующей таблицей:

П а ц и е н т ы	
РК	<u>Код пациента</u>
	Фамилия Имя Отчество Пол Тип пациента

**Рис. 1.** Таблица «Пациенты» – основной список

Здесь поле «Тип пациента» указывает, находится ли пациент в стационаре круглосуточно или только днем.

Для формирования параметров информационного блока создадим таблицу, в которой будем указывать все необходимые параметры – список возможных параметров. Эта таблица может изменяться пользователем в ходе эксплуатации ПК. Вид этой таблицы приведен на рис. 2.

П а р а м е т р ы	
РК	<u>Код параметра</u>
	Параметр Тип данных Способ ввода Таблица рабочая

**Рис. 2.** Таблица «Параметры» – таблица информационного блока

Назначение полей «Код параметра» и «Параметр» понятно из названия. Поле «Тип данных» определяет, к какому типу данных относятся значения данного параметра. Возможные значения: текстовый, целочисленный, вещественное число, дата, логический. Поле «Способ ввода» определяет, как пользователь буде вводить значения данного параметра. Возможные значения: ввод вручную, выбор единственного значения из списка, выбор нескольких значений из списка, единственного значения из списка + примечание (вводится вручную), выбор нескольких значений из списка + примечание (для каждого выбранного из списка значения). Скорее всего, способ ввода «выбор значения из списка» будет характерен для типа данных «текстовый». Таблица, в которой может быть сформирован список возможных значений для каждого параметра, показана на рис. 3 (для других типов данных также могут быть созданы аналогичные таблицы поле «Значение параметра» имеет тип, соответствующий полю «Тип данных» таблицы «Параметры»).

З н а ч е н и я   п а р а м е т р о в	
РК	<u>Код параметра</u>
РК	<u>Код значения</u>
	Номер п/п Значение параметра

**Рис. 3.** Таблица «Значения параметры» – таблица информационного блока

Поле «Таблица рабочая» (таблицы «Параметры») указывает таблицу, в которой будут храниться значения параметра, указанные для каждого пациента. Это поле связано с полем «Тип данных». Здесь предусмотрено 5 типов данных. Соответственно, должно быть 5 таблиц, в которых будут храниться значения параметров соответствующих типов. Каждый из параметров в этих таблицах будет связан с пациентами.

Поле «Номер п/п» (таблица «Значения параметров») используется для сортировки списка возможных значений параметра (для отображения пользователю при выборе требуемого значения). Эта таблица также может изменяться пользователем в ходе эксплуатации ПК (добавляться новые значения, изменяться и удаляться существующие).

Таблица, в которой будут храниться параметры определенного типа для каждого пациента, имеет следующую структуру.

П а р а м е т р ы п а ц и е н т а ( Т 1 )	
РК	<u>Код пациента</u>
РК	<u>Код параметра</u>
РК	<u>Код значения</u>
	Значение (тип Т1)

**Рис. 4.** Таблица «Параметры пациента» – таблица информационного блока

В приведенном примере поле «Значение» имеет тип «текстовый» (1-й тип из 5 возможных). Для хранения параметров, описывающих пациентов четырех оставшихся типов структура таблицы аналогичная, но поле «Значение» будет иметь тип соответственно целочисленный, вещественное число, дата, или логический. Образованы пять таблиц «Параметры пациентов» (каждая для своего типа) и таблицы «Параметры» и «Значения параметров». Совокупность этих семи таблиц полностью определяют информационный блок – блок параметров с описанием каждого пациента.

Отношения между таблицами, входящими в основной список и информационный блок показаны на рис. 5. Предполагается, что ввод значения параметра путем выбора требуемого значения из списка возможных

значений, используется только для типа данных «текстовый». Все остальные типы данных вводятся вручную (тип данных «текстовый» также может вводиться вручную). Поэтому таблица «Значения параметров» присутствует в единственном экземпляре (только для текстового типа данных). Если необходимо обеспечить выбор значения из списка для других типов, должны быть созданы аналогичные таблицы значений параметров соответствующих типов, которые будут связаны с таблицами «Параметры пациента», содержащих параметры этих типов.

Предлагаемая схема полностью отвечает первому требованию динамически настраиваемой ИС (изменение состава параметров). Изменение списка параметров не требует перепрограммирования интерфейса ПК. Интерфейс может программироваться как с использованием таких компонент, как таблицы, так и с помощью динамически создаваемых компонент визуализации и редактирования данных.

При этом обработчики событий, обеспечивающие редактирование соответствующих параметров (выбор значений из списка, удаление/замена значений), этих компонент носят универсальный характер.

Для определения состава журналов для данной ИС создадим таблицы, в которых будем регистрировать каждый новый журнал, с указанием какие поля, их типы и способ ввода данных будут входить в каждый журнал; а также таблицу для выбора значений для полей, способ ввода которых предполагает выбор значения из списка (как и прежде, полагаем, что выбор значения из списка может использоваться для типа данных «текстовый»). Структура и отношения этих таблиц показаны на рис. 5.

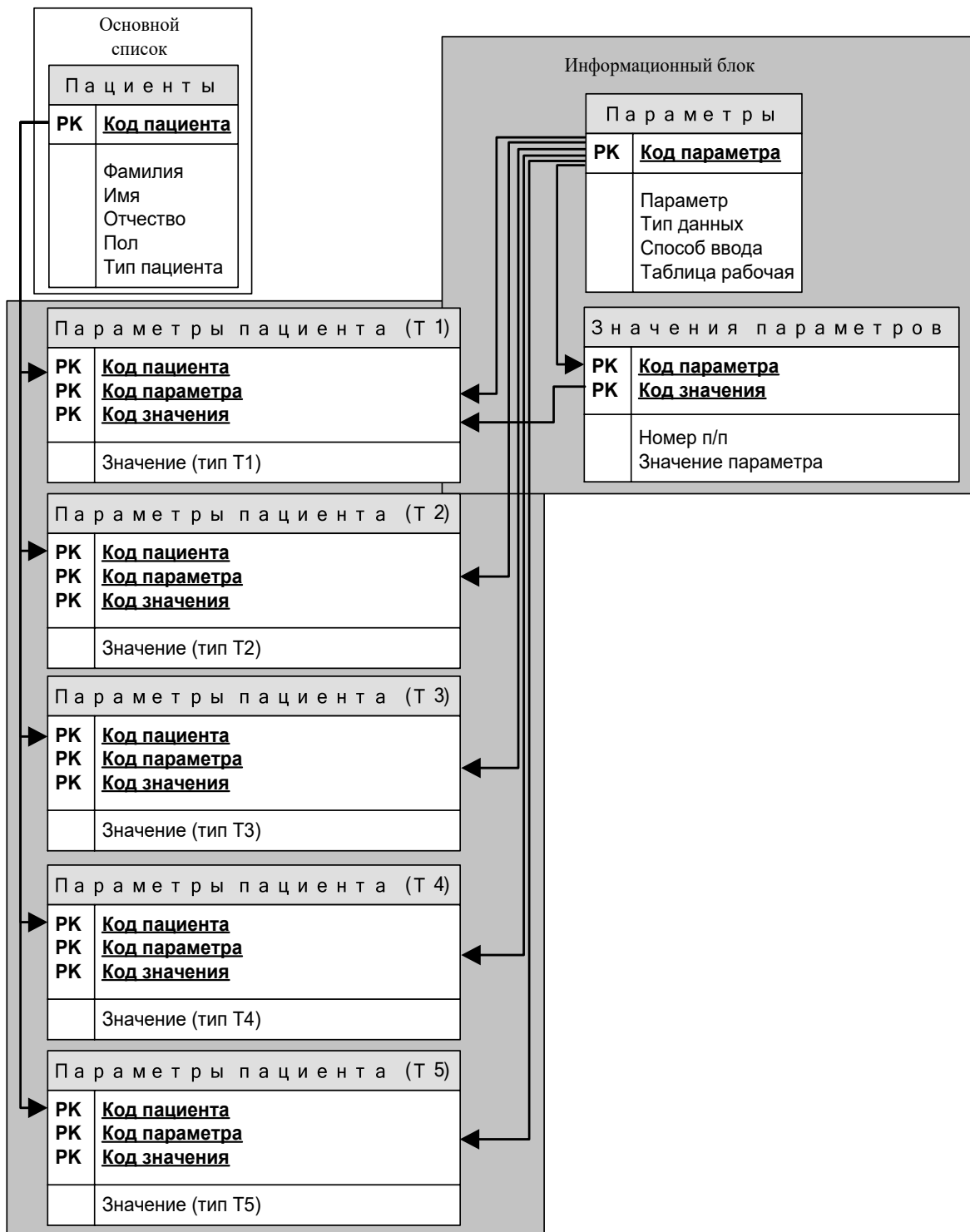
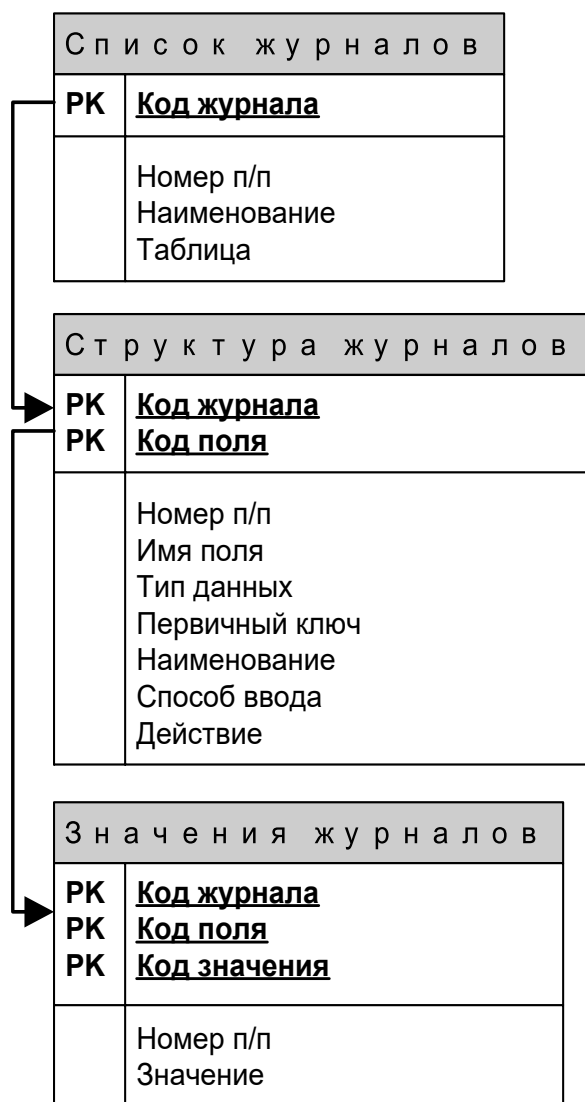


Рис. 5. Отношения между таблицами основного списка и информационного блока



**Рис. 6.** Таблицы регистрации создания журналов

Поле «Наименование» (таблица «Список журналов») используется для отображения названия журнала на закладке страниц (если в ИС журналов несколько, то логично каждый журнал располагать на отдельной странице). Поле «Таблица» (таблица «Список журналов») содержит собственно имя таблицы, в которой будет храниться информация данного журнала.

Поле «Имя поля» (таблица «Структура журналов») определяет имя поля, как оно будет указано в таблице данного журнала. Поле «Тип данных» определяет тип данных создаваемого поля. Поле «Первичный ключ» определяет, входит ли данное поле в состав первичного ключа создаваемого журнала (в состав первичного ключа журналов в обязательном порядке

будет входить как внешний ключ первичный ключ основного списка; полностью первичный ключ журналов образуется совокупностью первичного ключа основного списка и полей журнала, отмеченных, как входящих в первичный ключ). Поле «Наименование» используется для отображения соответствующей колонки в программе, работающей с данной базой данных. Поле «Способ ввода» аналогично такому же полю таблицы «Параметры». Поле «Действие» определяет какие действия необходимо выполнить при изменении (изменении значения, удалении значения) данного поля. В этом поле указывается имя метода (и при необходимости, имя программного компонента – dll-модуля, СОМ-сервера), который необходимо выполнить при изменении значения этого поля. Этим самым обеспечивается обработка событий при работе с компонентами ИС (третье требование динамически настраиваемых ИС). Поле «Номер п/п» во всех таблицах используется для сортировки записей при отображении информации. Таблица «Значения журналов» по своему назначению полностью идентична таблице «Значения параметров». На базе информации, хранящейся в этих системных таблицах, создаются таблицы-журналы, которые собственно и используются для хранения информации. Структура этих таблиц в обобщенном виде может быть представлена следующим образом.

Журнал (таблица)	
РК	<u>Код пациента</u>
РК	<u>&lt;Поля первичного ключа&gt;</u>
	<Прочие поля>

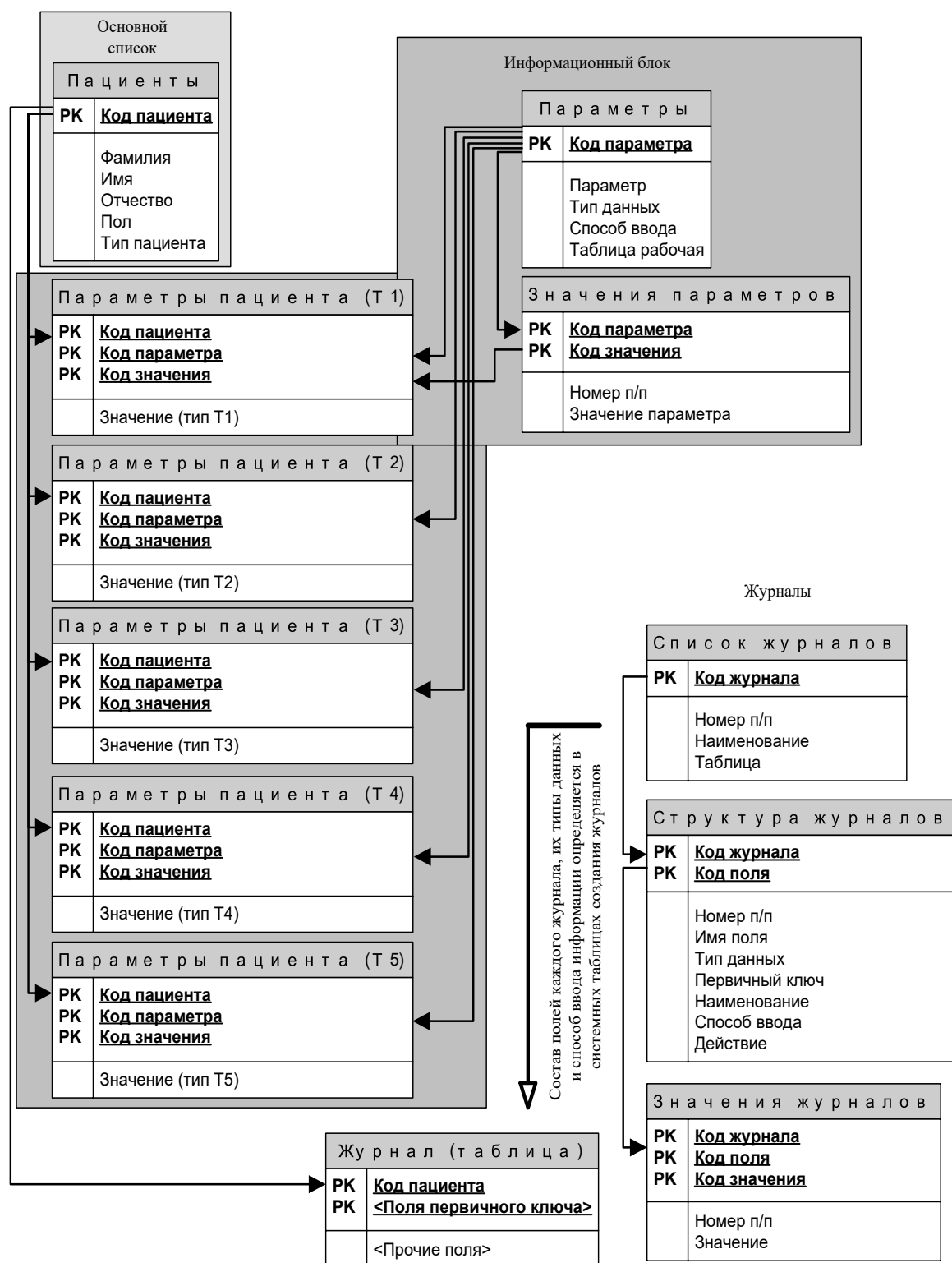
**Рис. 7.** Таблица-журнал – обобщенная структура

Эта таблица связана с основным списком (в данном примере по полю «Код пациента») и может быть связана с таблицей «Значения журналов», если какие-либо поля предполагают выбор значений из списка.

Состав и отношения между таблицами базы данных динамически настраиваемой ИС показана на рисунке ниже. На рисунке показан один журнал, хотя в реальной ИС их может быть несколько.

Предлагаемая схема организации данных полностью соответствует требованиям динамически настраиваемых ИС. При этом интерфейсная часть ПК не требует перепрограммирования при изменении состава параметров и/или журналов. В данном случае интерфейсная часть носит универсальный характер. Возможность создания собственных и подключение созданных другими сторонами обработчиков событий (с помощью dll-библиотек, или с помощью СОМ-серверов) позволяет проводить более тонкую настройку ИС. При этом не затрагивается ранее созданная программная составляющая ИС. Добавление в ПК функций СОМ-сервера и организация программного интерфейса, включающего доступ к методам обработки событий, позволит создавать программные объекты на базе объектно-ориентированного подхода. При этом вновь создаваемые обработчики событий могут иметь доступ к исходным («родительским») обработчикам событий ИС.





**Рис. 8.** Состав и отношения между таблицами базы данных динамически настраиваемой ИС

## УПРАВЛЕНИЕ ПРОЦЕССОМ ПРОТИВОТОЧНОГО ВЫЩЕЛАЧИВАНИЯ ПУЛЬСАЦИЯМИ РАСТВОРИТЕЛЯ

Анализируя полученные математические зависимости и экспериментальные данные по кинетике массо- и теплообмена, гидродинамики плотно-пульсирующего слоя зернистого полидисперсного материала, можно сформулировать основные задачи и критерий управления процессом выщелачивания бокситовых спёков.

Задача управления процессом выщелачивания состоит в стабилизации химического состава продукционного алюминатного раствора. Содержание полезных компонентов в продукционном растворе в реальном масштабе времени измерить сложно. Поэтому учитывая, что процесс выщелачивания происходит при выделении тепла от протекания экзотермических реакций, то учитывая это обстоятельство и зная тепловой эффект выщелачивания бокситовых спёков, задачу стабилизации химического состава продукционного алюминатного раствора можно рассматривать как задачу стабилизации температурного режима в колонном аппарате. Кроме того температуры по высоте колонны измерить довольно просто.

В условиях случайных возмущений нагрузки аппарата и физических свойств твёрдого зернистого материала, имеет место не стационарность режима и стохастические нарушения гидродинамики. Оперативная идентификация и ликвидация нарушений, как температурного, так и гидродинамического, осуществляется по математической модели.

Рассмотрим температурную модель процесса выщелачивания

$$\frac{\partial T_1}{\partial t} + V \frac{\partial T_1}{\partial Z} = D_p \frac{\partial^2 T_1}{\partial Z^2} - \frac{(1-\varepsilon)\lambda_1}{\varepsilon\rho_1 C_{p1}} (3\overline{M}_0 \varphi_1^2 \frac{\partial \varphi_1}{\partial t} + \frac{\kappa \overline{M}_0 \rho_s C_{ps}}{\rho} \frac{\partial \overline{T}_s}{\partial t}), \quad (1)$$

в случае когда  $\frac{\partial \overline{T}_s}{\partial t} = 0$ .

Тогда уравнение (1) запишется в виде

$$\frac{\partial T_1}{\partial t} + V \frac{\partial T_1}{\partial Z} = D_p \frac{\partial^2 T_1}{\partial Z^2} - \frac{(1-\varepsilon)}{\varepsilon} p \varphi_1^2 \frac{\partial \varphi_1}{\partial t}, \quad (2)$$

где  $p = \frac{4\lambda_1 M_0}{\rho_1 C_{p1} \pi R^3}$ .

Математическая модель (2) позволяет осуществить оперативную оценку распределения параметров и состояние объекта управления процесса выщелачивания по температурам  $T_1(z)$  по высоте аппарата. Подадим в колонну выщелачивающую жидкость температурой  $T_g$  со скоростью  $V$ , и длительностью по времени  $\tau_t$ , затем сделаем паузу  $\tau_p$  в подаче жидкости. Тогда уравнения (2) в разностном виде по времени  $t$  при  $V \neq 0$  запишем в виде

$$\frac{T_1(\tau_t) - T_1(0)}{\tau_t} + V \frac{\partial T_1(0)}{\partial Z} = D_p \frac{\partial^2 T_1(0)}{\partial Z^2} - \frac{(1-\varepsilon)}{\varepsilon} p \varphi_1^2(0) \frac{\varphi_1(\tau_t) - \varphi_1(0)}{\tau_t}. \quad (3)$$

Аналогично запишем уравнение в момент паузы в подаче выщелачивающей жидкости

$$\frac{T_1(\tau_t + \tau_p) - T_1(\tau_t)}{\tau_p} + V \frac{\partial T_1(0)}{\partial Z} = D_p \frac{\partial^2 T_1(0)}{\partial Z^2} - \frac{(1-\varepsilon_0)}{\varepsilon_0} p \varphi_1^2(\tau_t) \frac{\varphi_1(\tau_t + \tau_p) - \varphi_1(\tau_t)}{\tau_p}, \quad (4)$$

где  $\varepsilon_0$  – порозность слоя при  $V=0$ .

Умножим уравнение (3) на  $\tau_t$ , уравнение (4) на  $\tau_p$ . Получим

$$T_1(\tau_t) - T_1(0) + \tau_t V \frac{\partial T_1(0)}{\partial Z} = \tau_t D_p \frac{\partial^2 T_1(0)}{\partial Z^2} - \frac{(1-\varepsilon)}{\varepsilon} p \varphi_1^2(0) (\varphi_1(\tau_t) - \varphi_1(0)), \quad (5)$$

$$T_1(\tau_t + \tau_p) - T_1(\tau_t) = \tau_p D_p \frac{\partial^2 T_1(0)}{\partial Z^2} - \frac{(1-\varepsilon)}{\varepsilon} p \varphi_1^2(\tau_t) (\varphi_1(\tau_t + \tau_p) - \varphi_1(\tau_t)). \quad (6)$$

Сложим уравнения (5) и (6), получим

$$T_1(\tau_t + \tau_p) - T_1(0) + \tau_t V \frac{\partial T_1(0)}{\partial Z} = (\tau_t + \tau_p) D_p \frac{\partial^2 T_1(0)}{\partial Z^2} + \frac{(1-\varepsilon)}{\varepsilon} p K_1 + \frac{(1-\varepsilon_0)}{\varepsilon_0} p K_2, \quad (7)$$

где

$$K_1 = \varphi_1^2(0) (\varphi_1(0) - \varphi_1(\tau_t)),$$

$$K_2 = \varphi_1^2(\tau_t)(\varphi_1(\tau_t) - \varphi_1(\tau_t + \tau_p)).$$

Исходя из балансовых и гидродинамических соотношений, для псевдооживленного слоя можно написать

$$V\tau_t = V_s(\tau_t + \tau_p), V_s \geq W_p, \varepsilon \geq \varepsilon_0, \quad (8)$$

для плотно-пульсирующего слоя

$$V\tau_t = V_s(\tau_t + \tau_p), V_s \leq W_p, \varepsilon = \varepsilon_0, \quad (9)$$

где  $V_s$  – средняя скорость жидкости в колонне, соответствующая среднему расходу выщелачивающей жидкости в колонный аппарат за период  $T_p$ ,  $W_p$  – скорость псевдооживления.

Учитывая (8) и (9), соотношение (7) можно записать в виде

$$T_p D_p \frac{\partial^2 T_1(0)}{\partial Z^2} - T_p V_s \frac{\partial T_1(0)}{\partial Z} = T_1^{n+1} - T_1^n - f, \quad (10)$$

где  $f = p(\frac{1-\varepsilon}{\varepsilon} K_1 + \frac{1-\varepsilon_0}{\varepsilon_0} K_2)$ ,  $T_1^{n+1}$  – температура раствора по окончании периода  $T_p = \tau_t + \tau_p$ ,  $T_1^n$  – начальная температура в растворе для периода  $T_p$ .

Разбив колонный аппарат, по высоте, на ряд зон и записав уравнение (10) в виде разностных уравнений, получим

$$T_p D_p \frac{T_{li+1}^n - 2T_{li}^n + T_{li-1}^n}{h^2} - T_p V_s \frac{T_{li+1}^n - T_{li}^n}{h} = T_{li}^{n+1} - T_{li}^n - f_i. \quad (11)$$

Из (11) можно определить температуру раствора по окончании периода  $T_p$

$$T_{li}^{n+1} = T_{li}^n + T_p D_p \frac{T_{li+1}^n - 2T_{li}^n + T_{li-1}^n}{h^2} - T_p V_s \frac{T_{li+1}^n - T_{li}^n}{h} - f_i, \quad (12)$$

где  $T_{li}^n, T_{li}^{n-1}$  – измеряемые температуры в зоне  $i$  в текущий период и в предыдущий.

Для достижения заданного значения температуры в зоне  $i$  по высоте колонны, построим алгоритм управления, формирующей управляющее воздействие  $\tau_t$  – продолжительность подачи выщелачивающей жидкости по модели (12).

Исследуем функцию  $f_{li}(\tau_i)$ , входящую в выражение (12).

Рассмотрим плотно-пульсирующий слой, т.е. когда  $V_s \leq W_p$  – скорость псевдооживления, тогда  $\varepsilon = \varepsilon_0$ . Следовательно, в этом случае

$$f_{li}(\tau_i) = p \frac{1-\varepsilon}{\varepsilon} (\varphi_{li}^2(0)(\varphi_{li}(0) - \varphi_{li}(\tau_i)) + \varphi_{li}^2(\tau_i)(\varphi_{li}(\tau_i) - \varphi_{li}(\tau_i + \tau_p))).$$

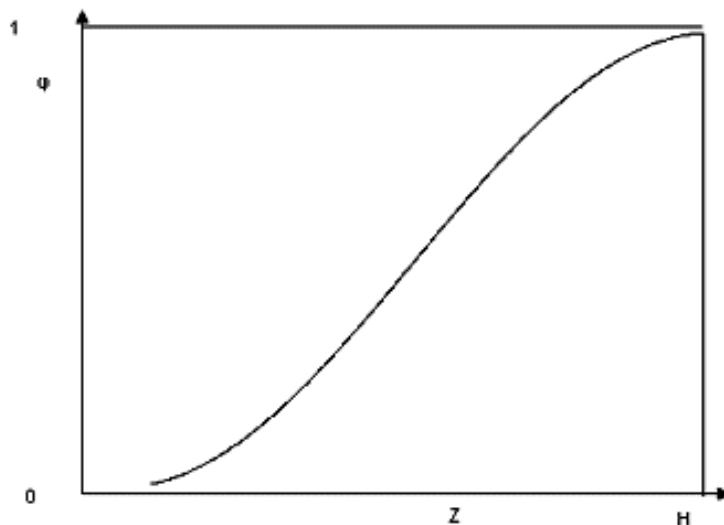
Для определения  $f_{li}(\tau_i)$  необходимо знать распределение  $\varphi_{li}(0)$  по высоте колонны. Для этого воспользуемся соотношением [2]

$$\frac{Dt_i}{R^2} = \left(\frac{1}{\gamma} + \frac{1}{2}(v-1)\right) \left(\frac{1}{6} - \frac{\varphi_{li}^2}{2} + \frac{\varphi_{li}^3}{3}\right) - \frac{1}{6}(\ln \varphi_{li} - \varphi_{li} + 1),$$

где  $t_i = \frac{H - Z_i}{W}$ .

Из последнего соотношения для каждой зоны  $i$  находим  $\varphi_{li}(0)$ . Общий вид зависимости  $\varphi_{li}$  от высоты колонны и  $f_{li}(\tau_i)$  от продолжительности подачи жидкости в аппарат представлены на рис. 1 и рис. 2.

Таким образом, задача управления режимом выщелачивания сводится к решению уравнения (12) на каждом такте подготовки управляющего воздействия относительно  $\tau_i$ . Для каждой зоны  $i$  по высоте колонного аппарата получим своё значение импульса  $\tau_{ti}$ .



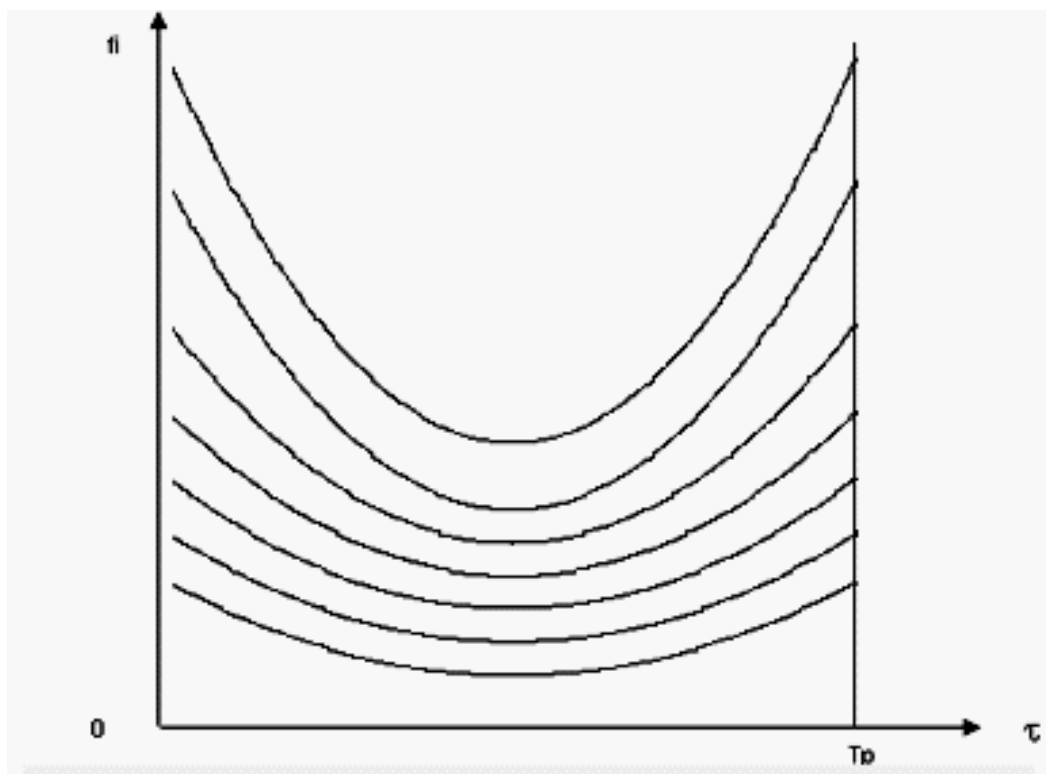
**Рис.1.** Зависимость безразмерного радиуса от координаты  $Z$  по высоте колонны

Если решений уравнения (12) для  $\tau_i \in [0, T_p]$  нет, то берётся значение функции  $f_{i_i}(\tau_i)$  в точке, когда она достигает своего экстремума, т.е. в точке  $\max[f_{i_i}(0), f_{i_i}(T)]$  или в точке  $\min[f_{i_i}(\tau_i)]$  в зависимости от отклонения текущей температуры от заданной.

Время и длительность периода подачи выщелачивающей жидкости в колонный аппарат определим как

$$\tau_i = \sum_{i=1}^n K_i \tau_{ti}, T_p = \sum_{i=1}^n K_i T_{pi},$$

где  $n$  – количество рассматриваемых зон выщелачивания по высоте колонного аппарата,  $K_i$  – нормирующие коэффициенты,  $\sum_{i=1}^n K_i = 1$ .



**Рис. 2.** Зависимость функции  $f_i$  от времени подачи жидкости на выщелачивание

## **ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ – ОРГАНИЗАТОР И КАТАЛИЗАТОР ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА**

До сих пор массовое обучение, ориентированное на «определенную» специальность доминирует в вузах страны. Когда стране не хватало специалистов, этот тип обучения, безусловно, оправдал себя. Кроме того, уровень производства и управления в те времена не требовал многочисленных специализаций. Студенту надо было передать определенный объем знаний и сформировать у него необходимые умения и навыки. Требуемый объем базовой подготовки определялся государственными стандартами и вузом. Было время, когда считали, что вузовской подготовки хватит выпускнику лет на десять его производственной деятельности. В настоящее время ситуация изменилась коренным образом – выпускнику надо пополнять свои знания сразу после окончания вуза, так как существенно возросла наукоемкость и инновационная интенсивность в производстве и управлении. Кроме того, многократно увеличилось количество специализаций, по которым требуются специалисты. С другой стороны, обеспечение высокого уровня квалификации предполагает постоянное обучение на протяжении всего периода профессиональной деятельности. Традиционное повышение квалификации с отрывом от производственной деятельности на несколько месяцев все чаще становится обременительным для небольших фирм и самих специалистов. Обычно, только крупные фирмы могут позволить себе такую форму повышения квалификации своих сотрудников. Тем более что со временем, потребность в получении разнообразных систематизированных знаний по конкретным вопросам будет возрастать.

Следует отметить, что внедряемая в настоящее время в вузах страны технология двухуровневой подготовки специалистов, также не снимает главной проблемы – каким образом готовить профессионалов в конкретной предметной области, отвечающих современным требованиям. Да, можно за четыре года подготовить бакалавра с хорошей теоретической подготовкой. Но это не специалист, имеющий представление о специфике конкретной предметной области. Возникает главный вопрос: а можно ли

вообще сформировать специалиста, хотя бы в базовом варианте, за один год? Другой вопрос: как обеспечить постоянно возрастающее число специализаций? Общий вопрос: каким образом специалист может оперативно получать необходимые сведения из стремительно разрастающихся информационных ресурсов? Ответ на перечисленные и другие многочисленные вопросы один – внедрение в вузы комплекса информационных технологий. В настоящее время этот процесс неуправляемый, и решается вузами в соответствии с их ресурсными возможностями. К основным функциям вуза, с точки зрения информационного обеспечения, следует отнести:

- общее администрирование (финансы, кадры и организация эксплуатации основных фондов);
- сопровождение информационных ресурсов вуза всех типов;
- управление, обеспечение, мониторинг и контроль качества учебного процесса;
- информационное обеспечение научно-исследовательской работы;
- информационное обеспечение центров коллективного пользования и технологических центров;
- обеспечение многочисленных и многообразных связей вуза с внешним миром.

Самостоятельная даже частичная реализация вузами перечисленных функций приводит к неэффективному расходованию их ограниченных ресурсов. К сожалению, эта серьезная проблема сейчас вне круга интересов соответствующих государственных структур.

Очевидно, что любая технология, в том числе и технология образовательного процесса, соответствует ресурсным возможностям своего времени. Появление новых информационных технологий представления знаний, стремительная миниатюризация технических средств обработки данных, резкое увеличение коммуникационных возможностей обуславливают формирование новой методологии образования, в разработку и распространение которой необходимо уже сейчас вкладывать значительные средства. Если решение этой важнейшей проблемы отложить на неопределенное время, то уже совсем скоро отработанная за многие годы традиционная система организации образовательного процесса превратится в настоящий тормоз непрерывного образования.



На протяжении многих лет основным источником знаний для большей части студентов был преподаватель. Важную роль играли также учебники, учебные пособия и учебно-методические указания. Основным источником справочных сведений была библиотека. Поэтому пребывание в вузе было главным фактором получения качественного образования.

Широкое распространение гипертекстовой технологии и Интернета, мультимедийные возможности телеконференций в реальном времени коренным образом изменили структуру и состояние мирового информационного пространства. Революционное значение внедрения в образование гипертекстовой технологии заключается в многообразии используемых объектов: текстовых, аудио и видео. Если ранее базовым в образовании было широкое использование логического мышления (функция левого полушария мозга), а образное мышление (функция правого полушария мозга) использовалось незначительно, то в настоящее время ситуация меняется кардинально. Применение в гипертексте графики, компьютерной анимации и видео позволяет применять в образовании образное мышление на полную мощь. При этом существенно возрастает качество восприятия новых знаний, за счет большего многообразия эмоций.

Безусловно, гипертекст повышает активность обучаемого в процессе в процессе формирования у него новых знаний. Использование гипертекста – важный шаг в решении главной проблемы образования – обеспечения индивидуального интерактивного обучения. К основным задачам этой проблемы можно отнести:

- разработку методов и средств создания индивидуальной активной обучающей среды;
- формирование способов индивидуального погружения в активную обучающую среду.

Одним из средств создания обучающей среды нового типа является FLASH-технология. Ответ на другой вопрос должны дать психологи. У одного обучаемого предлагаемый вариант взаимодействия с обучающей средой вызывает удовлетворение и ощущение комфорта, а у другого неприязнь, или, более того, раздражение. Одно ясно, эта парадигма принципиально изменяет суть и характер учебного процесса. Обучаемый не просто воспринимает знания, а становится активным участником их приобретения.

Переход к новой парадигме образования предполагает серьёзную перестройку всего образовательного процесса. Механическое преобразование традиционного (линейного) текста учебных материалов в стандартный гипертекст обычно сопровождается незначительным эффектом. Серьёзные результаты можно получить только в результате принципиальной переработки учебного материала. Важным результатом включения информационных технологий в учебный процесс является глубокая детализация содержания учебных курсов. Необходимо разделение содержания учебного курса на множество взаимосвязанных частей различного объема и природы, между которыми могут быть установлены сотни связей. При этом на первый план выходит семантическая структура учебного курса:

1. Представление о предметной области изучаемой дисциплины.
2. Основные понятия: их содержание и объём.
3. Назначение и роль основных изучаемых процессов.
4. Главные объекты изучаемых процессов: их свойства и реализуемые методы, характер и способы взаимодействия объектов.
5. Основные законы изучаемой предметной области.
6. Типовые модели и задачи изучаемой предметной области.
7. Типовые методы и способы решения задач.

Внедрение в сознание студентов многочисленных связей между разделами курса в традиционном изложении требует серьёзных усилий лектора. В то время как в гипертексте связи устанавливаются естественным образом. Применение информационных технологий радикально изменяет способы изучения учебной дисциплины. В традиционном изложении у большей части студентов в распоряжении был только один сценарий изучения курса – избранный преподавателем. В то время как интерактивный диалог допускает применение множества сценариев, которые могут изменяться в процессе изучения курса, с тем, чтобы максимальной степени соответствовать потребностям обучаемого. Активная обучающая среда может быть реализована только в электронном виде, поэтому для её создания требуются значительные ресурсы. Но зато она может храниться на компактном носителе, легко тиражируется и передается на любые расстояния.

Качество обучающей среды целесообразно оценивать группой независимых экспертов, мнение которых должно быть решающим при государственной регистрации.

## **МЕТОДИКА ПРОЕКТИРОВАНИЯ ОХРАННОЙ СИСТЕМЫ ЗАЩИТЫ КОМПЬЮТЕРНОЙ СИСТЕМЫ ПРЕДПРИЯТИЯ**

**Концептуальная модель охранной системы защиты предприятия (компьютерной системы – КС).** Систему защиты КС предприятия можно рассматривать как сложную систему, включающую в себя некоторое множество элементов средств защиты и множество возможных каналов несанкционированного доступа, которая решает не только задачу защиты информационных ресурсов, но и обеспечения необходимого уровня информативности информационных процессов, происходящих внутри системы предприятия и во внешней среде по отношению к ней.

Система защиты КС предприятия должна удовлетворять следующим требованиям:

- быть экономически оправданной (т.е. иметь технико-экономическое обоснование);
- быть достаточно простой в эксплуатации. Не создавать дополнительных трудностей в работе предприятия;
- обеспечивать защиту всех реально возможных угроз.

При организации системы защиты необходимо сформировать цель, представить эту систему в виде совокупности простых, частных элементов средств защиты, установить отношения между этими элементами, например с помощью графа-редукции.

Вершинам графа  $G(R)$  ставят в соответствие элементы множества  $X$ , и если  $x_i \in R x_j$ , то от вершины  $x_i$  проводят ориентированную дугу к вершине  $x_j$ ; если  $x_i \notin R x_j$ , то дуга отсутствует.

Пусть элементами системы охранной защиты КС предприятия являются:

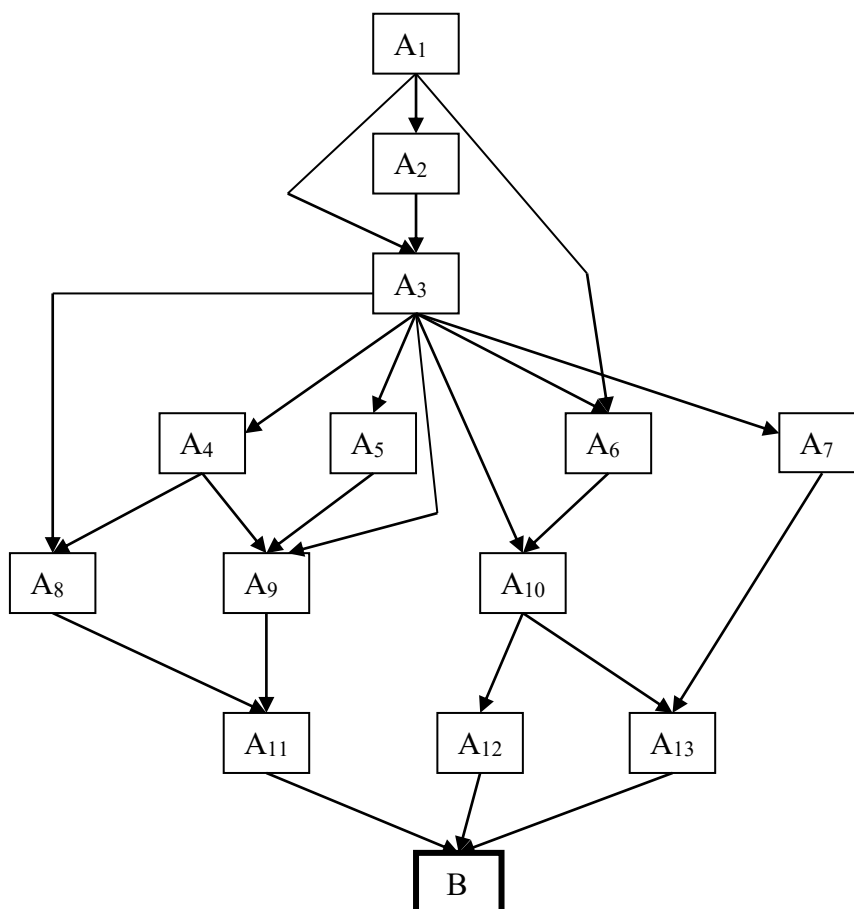
$A_1$  – первый контур защиты – средства контроля доступа на территорию объекта,

$A_2$  – второй контур защиты – средства контроля и разграничения доступа в помещения,

$A_3$  – третий контур защиты – основной ( $A_4 – A_{13}$ ),

$A_4$  – средства опознавания и разграничения доступа к информации.  
 $A_5$  – средства контроля вскрытия аппаратуры,  
 $A_6$  – каналные соединения,  
 $A_7$  – средства регистрации и учета документации,  
 $A_8$  – средства контроля доступа к терминалам,  
 $A_9$  – средства контроля программного обеспечения,  
 $A_{10}$  – средства контроля доступа к носителям информации,  
 $A_{11}$  – средства контроля вывода аппаратуры из процесса обмена информацией,  
 $A_{12}$  – средства контроля доступа к секретной информации (шифрование),  
 $A_{13}$  – средства контроля доступа к уничтожаемым носителям информации,  
 $B$  – выход из охраняемого объекта.

Тогда концептуальную модель системы защиты КС можно представить в виде рис. 1.



**Рис. 1.** Концептуальная модель системы защиты

Метод экономического расчета затрат на проектируемую охранную систему защиты информации (ОСЗИ). Этот метод расчета затрат состоит из двух этапов:

- расчет возможного денежного ущерба ( $C_k$ ) от утечки и разрушения информации,
- экономический расчет распределения затрат на проектируемую охранную систему защиты информации.

Оценка ценности информационных ресурсов должна производиться экспертами на основе определения стоимостных (в смысле ущерба от утечки) характеристик информации, которая может «утечь» или разрушиться через каждый возможный канал утечки с использованием метода «худшего случая».

Каждый эксперт дает оценку ущерба предприятия (в рублях), которая может получиться в результате утечки или разрушения каждого из сведений, подлежащих защите.

При проведении оценки эксперты должны дать ответы на следующие вопросы:

1. Каков прямой денежный ущерб ( $C_{k1}$ ) от утечки данной информации?
2. Каков денежный ущерб ( $C_{k2}$ ) от утраченной выгоды предприятия?
3. Каков денежный ущерб ( $C_{k3}$ ) от замедления дел предприятия?
4. Какие усилия будут необходимы, чтобы нейтрализовать причиненный ущерб ( $C_{k4}$ )?

Возможный денежный ущерб ( $C_k$ ) от утечки или разрушения каждого  $k$ -го сведения подсчитывается как сумма  $C_{ki}$

$$C_k = C_{k1} + C_{k2} + C_{k3} + C_{k4},$$

где  $k = 1, 2, \dots, n$  – число сведений, подлежащих защите.

В основу экономического расчета распределения затрат на проектируемую охранную систему защиты информации положена концептуальная модель (см. рис. 1).

При анализе модели системы защиты КС предприятия необходимо использовать положения системного анализа, согласно которым декомпозиция этой системы представлена схемой концептуальной модели. Поведение этой системы можно найти вычислением поведения наибольшего элемента разложения  $B$ , используя арифметико-логические связи:

$$A_j = \sum_{i=1}^n a_{ij} \times A_i,$$

где  $A_i$  – вершина, а  $a_{ij}$  – дуги графа.

Арифметико-логические связи нашей схемы (см. рис. 1) имеют вид:

$$\begin{aligned} A_2 &= a_{12} * A_1, A_3 = a_{23} * A_2 + a_{13} * A_1, A_4 = a_{34} * A_3, A_5 = a_{35} * A_3, \\ A_6 &= a_{36} * A_3 + a_{16} * A_1, A_7 = a_{37} * A_3, A_8 = a_{38} * A_3 + a_{48} * A_4, \\ A_9 &= a_{49} * A_4 + a_{59} * A_5, A_{10} = a_{310} * A_3 + a_{610} * A_6, \\ A_{11} &= a_{911} * A_9 + a_{311} * A_3 + a_{811} * A_8, A_{12} = a_{1012} * A_{10}, \\ A_{13} &= a_{1013} * A_{10} + a_{713} * A_7, \\ B &= a_{11} * B * A_{11} + a_{12} * B * A_{12} + a_{13} * B * A_{13}. \end{aligned}$$

Решая приведенные логические уравнения, находим пути достижения вершины  $B$ . Результаты вычисления сведены в матрицу смежности  $Q = [q_{ij}]_{m \times n}$  (табл. 1), каждому элементу которой взаимно однозначно соответствует элемент системы защиты, входящий в событие (путь), каждой строке  $S$  — совокупность элементов, при наличии которых событие (путь) имеет место (истинно):

$$Q = \begin{vmatrix} q_{11} & q_{12} & \dots & q_{1n} \\ q_{21} & q_{22} & \dots & q_{2n} \\ \dots & \dots & \dots & \dots \\ q_{m1} & q_{m2} & \dots & q_{mn} \end{vmatrix},$$

где  $q_{ij} = \begin{cases} 1, & \text{если } j\text{-й элемент входит в } i\text{-ю совокупность элементов,} \\ & \text{при наличии которых событие истинно,} \\ 0 & \text{— в противном случае.} \end{cases}$

Для определения узких мест в системе защиты, т.е. поиск элементов защиты, находящихся под наибольшим напряжением угрозы, необходимо определить интенсивность участия элементов защиты в путях несанкционированного доступа к информации. Интенсивность участия элементов в событии можно найти, используя матрицу частотных отношений  $F = (w_{ij})_{n \times n}$ .

Матрица смежности  $Q$ 

$S \backslash A$	A <sub>1</sub>	A <sub>2</sub>	A <sub>3</sub>	A <sub>4</sub>	A <sub>5</sub>	A <sub>6</sub>	A <sub>7</sub>	A <sub>8</sub>	A <sub>9</sub>	A <sub>10</sub>	A <sub>11</sub>	A <sub>12</sub>	A <sub>13</sub>	B
S <sub>1</sub>	1	1	1					1			1			1
S <sub>2</sub>	1		1					1			1			1
S <sub>3</sub>	1		1			1					1			1
S <sub>4</sub>	1		1	1				1			1			1
S <sub>5</sub>	1	1	1	1				1			1			1
S <sub>6</sub>	1	1	1	1					1		1			1
S <sub>7</sub>	1		1	1					1		1			1
S <sub>8</sub>	1	1	1		1				1		1			1
S <sub>9</sub>	1		1		1				1		1			1
S <sub>10</sub>	1	1	1								1			1
S <sub>11</sub>	1		1								1			1
S <sub>12</sub>	1	1	1							1		1		1
S <sub>13</sub>	1	1								1		1		1
S <sub>14</sub>	1	1	1			1				1		1		1
S <sub>15</sub>	1		1			1				1		1		1
S <sub>16</sub>	1					1				1		1		1
S <sub>17</sub>	1	1	1							1			1	1
S <sub>18</sub>	1		1							1			1	1
S <sub>19</sub>	1	1	1			1				1			1	1
S <sub>20</sub>	1		1			1				1			1	1
S <sub>21</sub>	1					1				1			1	1
S <sub>22</sub>	1	1	1				1						1	1
S <sub>23</sub>	1		1				1						1	1

Частотная матрица отношений  $F$ , матрица смежности которой  $Q$ , имеет вид (табл. 2):

$$F = Q^T \times Q = \begin{pmatrix} w_{11} & w_{12} & w_{13} & \dots & w_{1n} \\ w_{21} & w_{22} & w_{23} & \dots & w_{2n} \\ \dots & \dots & \dots & \dots & \dots \\ w_{n1} & w_{n2} & w_{n3} & \dots & w_{nn} \end{pmatrix}$$

где  $Q^T$  – матрица транспонированная (табл. 3).

Таблица 2

Частотная матрица отношений  $F$ 

$A \backslash A$	$A_1$	$A_2$	$A_3$	$A_4$	$A_5$	$A_6$	$A_7$	$A_8$	$A_9$	$A_{10}$	$A_{11}$	$A_{12}$	$A_{13}$	$B$
$A_1$	23	11	20	4	2	7	2	4	4	10	11	5	7	23
$A_2$	11	11	10	2	1	3	1	2	2	5	5	3	3	11
$A_3$	18	9	18	4	2	5	1	4	4	6	11	2	5	18
$A_4$	4	2	4	4	0	0	0	2	2	0	4	0	0	4
$A_5$	2	1	2	0	2	0	0	0	2	0	2	0	0	2
$A_6$	7	1	5	0	0	7	0	0	0	6	1	3	3	7
$A_7$	2	1	2	0	0	0	2	0	0	0	0	0	2	2
$A_8$	4	2	4	2	0	0	0	4	0	0	4	0	0	4
$A_9$	4	2	4	2	2	0	0	0	4	0	4	0	0	4
$A_{10}$	10	4	7	0	0	6	0	0	0	10	0	5	5	10
$A_{11}$	11	4	11	4	2	1	0	4	4	0	11	0	0	11
$A_{12}$	5	3	3	0	0	3	0	0	0	5	0	5	0	5
$A_{13}$	7	3	6	0	0	3	2	0	0	5	0	0	7	7
$B$	23	11	20	4	2	7	2	4	4	10	11	5	7	23

Таблица 3

Транспонированная матрица  $Q^T$ 

$A \backslash S$	$S_1$	$S_2$	$S_3$	$S_4$	$S_5$	$S_6$	$S_7$	$S_8$	$S_9$	$S_{10}$	$S_{11}$	$S_{12}$	$S_{13}$
$A_1$	1	1	1	1	1	1	1	1	1	1	1	1	1
$A_2$	1				1	1		1		1		1	1
$A_3$	1	1	1	1	1	1	1	1	1	1	1	1	
$A_4$				1	1	1	1						
$A_5$								1	1				
$A_6$			1										
$A_7$													
$A_8$	1	1		1	1								
$A_9$						1	1	1	1				
$A_{10}$												1	1
$A_{11}$	1	1	1	1	1	1	1	1	1	1	1		
$A_{12}$												1	1
$A_{13}$													
$B$	1	1	1	1	1	1	1	1	1	1	1	1	1



Продолжение табл. 3

A \ S	S <sub>14</sub>	S <sub>15</sub>	S <sub>16</sub>	S <sub>17</sub>	S <sub>18</sub>	S <sub>19</sub>	S <sub>20</sub>	S <sub>21</sub>	S <sub>22</sub>	S <sub>23</sub>
A <sub>1</sub>	1	1	1	1	1	1	1	1	1	1
A <sub>2</sub>	1			1		1			1	
A <sub>3</sub>	1	1		1	1	1	1		1	1
A <sub>4</sub>										
A <sub>5</sub>										
A <sub>6</sub>	1	1	1			1	1	1		
A <sub>7</sub>									1	1
A <sub>8</sub>										
A <sub>9</sub>										
A <sub>10</sub>	1	1	1	1	1	1	1	1		
A <sub>11</sub>										
A <sub>12</sub>	1	1	1							
A <sub>13</sub>				1	1	1	1	1	1	1
B	1	1	1	1	1	1	1	1	1	1

Для оценки путей  $S_i$  граф-схемы концептуальной модели вычислим веса направленных дуг (табл. 4).

Для оценки прочности путей  $P_{si}$  граф-схемы концептуальной модели определяются веса вероятных путей с помощью дифференцирования (табл. 5).

Ущерб, причиненный компьютерными преступлениями оценить практически очень сложно, а иногда и невозможно. Проблема определения ущерба обычно требует определенных усилий по вычислениям и вероятностным определениям.

Используя принципы диаграмм Хассе и модели Фогта, можно определить вероятности контролируемых каналов несанкционированного доступа по наиболее вероятным путям по формуле:

$$P_{si} = \left[ 1 - \prod_{\substack{i=1 \\ j=i+1}} (1 - p(a_{ij})) \right]$$

где  $P(a_{ij})$  – нормированные значения весов (вероятностей) дуг граф-схемы концептуальной модели (см. табл. 4).

Таблица 4

Таблица весов направленных дуг

	A <sub>1</sub>	A <sub>2</sub>	A <sub>3</sub>	A <sub>4</sub>	A <sub>5</sub>	A <sub>6</sub>	A <sub>7</sub>	A <sub>8</sub>	A <sub>9</sub>	A <sub>10</sub>	A <sub>11</sub>	A <sub>12</sub>	A <sub>13</sub>	B
A <sub>1</sub>														
A <sub>2</sub>	0,1													
A <sub>3</sub>	0,005	0,08												
A <sub>4</sub>	0,45	0,52	0,33											
A <sub>5</sub>	1	0,43	0,76	0										
A <sub>6</sub>	0,28	0,38	0,4	0	0									
A <sub>7</sub>	1	0,43	0,76	0	0	0								
A <sub>8</sub>	0,45	0,52	0,33	0,19	0	0	0							
A <sub>9</sub>	0,45	0,26	0,33	0,19	0,09	0	0	0						
A <sub>10</sub>	0,12	0,21	0,26	0	0	0,08	0	0	0					
A <sub>11</sub>	0,46	0,23	0,06	0,16	0,43	0,67	0	0,17	0,170	0				
A <sub>12</sub>	0,34	0,32	0,85	0	0	0,19	0	0	0	0,09	0			
A <sub>13</sub>	0,22	0,32	0,28	0	0	0,26	0,23	0	0	0,13	0	0		
B	0	0,1	0,028	0,45	1	0,22	1	0,45	0,4	0,98	0,7	0,34	0,27	

Таблица 5

Таблица оценки прочности путей

S <sub>i</sub>	S <sub>1</sub>	S <sub>2</sub>	S <sub>3</sub>	S <sub>4</sub>	S <sub>5</sub>	S <sub>6</sub>	S <sub>7</sub>	S <sub>8</sub>	S <sub>9</sub>	S <sub>10</sub>	S <sub>11</sub>	S <sub>12</sub>
P <sub>Si</sub>	0,58	0,5	0,85	0,66	0,62	0,66	0,65	0,86	0,84	0,29	0,22	0,64

S <sub>i</sub>	S <sub>13</sub>	S <sub>14</sub>	S <sub>15</sub>	S <sub>16</sub>	S <sub>17</sub>	S <sub>18</sub>	S <sub>19</sub>	S <sub>20</sub>	S <sub>21</sub>	S <sub>22</sub>	S <sub>23</sub>
P <sub>Si</sub>	0,59	0,7	0,69	0,6	0,61	0,57	0,71	0,68	0,58	0,89	0,87

Рассчитанные вероятности  $P_{Si}$  позволят более объективно спроектировать и организовать систему защиты информации. Для расчета распределения затрат на проектируемую охранную систему защиты используем метод максимального элемента.

Допустим, система защиты состоит из  $n$  последовательного соединения средств. Прочность (вероятность преодоления прочности в течение заданного промежутка времени)  $i$ -го элемента задана величиной  $P_i$ .

Требуется распределить  $S_3$  условных единиц для повышения прочности отдельных элементов так, чтобы суммарная прочность системы защиты была наибольшей.

Рассмотрим распределение затрат на элементы наиболее вероятного пути ( $S_{кр}$ ).

Пусть  $S_{кр}$  состоят из 5 элементов, соединенных последовательно. Вероятность (преодоления прочности, которую задаем в начале проектирования системы)  $P_i$  и стоимости  $C_i$  элементов даны в табл. 6. Требуется резервировать систему так, чтобы ее суммарная стоимость не превышала  $C_3 \geq C_k$ , а прочность была бы максимальной.

Принимаем  $C_3 = C_0 + C_{доп}$

Таблица 6

**Вероятности и стоимость элементов**

$i$	1	2	3	4	5
$P_i$	0,1	0,08	0,76	0,26	0,27
$C_i$	1	0,7	3	0,4	0,5

Прочность незарезервированного пути  $S_{кр}$  равна

$$F_0 = \prod_{i=1}^n (1 - P_i) = 0,11,$$

а начальная стоимость  $C_0 = \sum C_i = 5,6$  у.е.

Формула для вычисления прироста прочности элементов пути на условную единицу имеет вид:  $\alpha(k, i) = \frac{P_i^k (1 - P_i)}{C_i (1 - P_i^k)}$

Результаты расчета  $\alpha(k, i)$  представлены в табл. 7

Таблица 7

**Прирост прочности элементов пути на 1 у.е.**

$k / I \alpha(k, i)$	1	2	3	4	N=5
1	0,1000	0,1143	0,2533	0,6500	0,5400
2	0,0091	0,0085	0,1094	0,1341	0,1148
3	0,0001	0,0007	0,0626	0,0331	0,0293
4	0,0000	0,0001	0,0401	0,0085	0,0078
5	0,0000	0,0000	0,0272	0,0022	0,0021
6	0,0000	0,0000	0,0191	0,0006	0,0006
7	0,0000	0,0000	0,0137	0,0001	0,0002
$Y_i^o$	2	2	5	4	4

Выбирая из табл. 7 последовательно  $\alpha(k, i)_{\max}$ , получаем следующую цепочку назначений:

$$(l_i) = 4 \rightarrow 5 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 2 \rightarrow 1 \rightarrow 3 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 3.$$

Компоненты вектора  $Y_0 = \{y_i^0\}$  определяются как количество выбранных из каждого столбца элементов, увеличенное на единицу ( $y_i^0 + 1$ ). Суммарная прочность пути для полученного решения  $P_{22} = 0,997$ .

Последовательность нарастания дополнительной стоимости в соответствии с цепочкой назначения будет:

$$(C_{\text{доп}}) = 0,4 \rightarrow 0,5 \rightarrow 3 \rightarrow 0,4 \rightarrow 0,5 \rightarrow 0,7 \rightarrow 1 \rightarrow 3 \rightarrow 3 \rightarrow 0,4 \rightarrow 0,5 \rightarrow 3 = 16,4.$$

Полная стоимость элементов  $S_{\text{кр}}$  равна:

$$C_3 = C_0 + C_{\text{доп}} = 5,6 + 16,4 = 22 \text{ у.е.}$$

Расчет для остальных элементов системы защиты, которые входят в пути  $S_5$ ,  $S_8$  и  $S_{14}$ , за исключением элементов, входящих в  $S_{\text{кр}}$ , т. е.  $S_{22}$ , можно выполнить аналогично.

### Библиографический список

1. *Мафтик С.* Механизм защиты в сетях ЭВМ: Пер с англ. М.: Мир, 1993. 216 с., ил.
2. *Панфилов И. В., Половко А. М.* Вычислительные системы / Под ред. А. М. Половко. М.: Советское радио, 1980. 304 с., ил.
3. *Панфилова Е. И.* Проблема оценки проектируемой системы защиты информации от компьютерных преступлений // В Сб. «Труды Санкт-Петербургского юридического института Генеральной прокуратуры Российской Федерации» № 2. СПб, 2000. С. 113–120.
4. *Березин Е. А.* Оптимальное распределение ресурсов и элементы синтеза систем / Под ред. Е. В. Зологова. М.: Советское радио, 1974. 304 с.

**В. А. Пресняков**, кандидат технических наук, доцент

## ТЕОРИЯ И ПРАКТИКА БД В СПбГЛТА

Направление СУБД получило развитие на кафедре в конце 80-х годов. Начало работ было связано с DBASE. Интересно отметить начальные вехи работы инициативной группы:

- В. Горбачев и С. Хабаров разрабатывают универсальную систему запросов;
- 1990 – В. Пресняков, С. Гуров, С. Хабаров разрабатывают автоматизированные рабочие места для спичечной фабрики «Пролетарское Знамя»;
- В. Горбачев – отлаживает систему планирования и учета для завода абразивных кругов, а также систему планирования договоров в учебных заведениях;
- 1991 – на кафедре положено начало разработкам по построению комплексных информационных систем по сбору производственной информации, учету производства и реализации продукции на предприятиях лесной и деревообрабатывающей промышленности (В. Пресняков, С. Хабаров, В. Лебедев и др.);
- 1991 началась работа по разработке информационной системы для первого совместно российско-финского предприятия «Чудово-RWS». Программные продукты, в том числе система учета и реализации **ФАСПР** (руководитель В. Пресняков) была внедрены на данном предприятии и успешно эксплуатировалась более 15 лет.

В первой половине 90-х на кафедре была построена первая локальная сеть на базе учебного класса. В учебный процесс по информационным технологиям и сетевым решениям был включен программный продукт **ФАСПР**, который позволил студентам получать представление о работе в сети с базой данных в многопользовательском режиме.

В это же самое время продолжалось развитие направления по проектированию и созданию больших лесосушильных камер аэродинамического типа (автор и изобретатель Смирнов Алексей Александрович). Игорь Борисович Шубинский поставил задачу о промышленном производстве такой продукции. Была разработана конструкторская документация, на основе

которой был размещен заказ на производство нескольких сушильных камер на вагоностроительном заводе в г. Тверь. Одна из камер была поставлена в технопарк ЛТА, где использовалась для сушки древесины.

С начала 2000 на основе огромного опыта, накопленного сотрудниками кафедры в сфере построения СУБД, а также знаний в специфике производственных вопросах и построению информационных систем для предприятий лесной и деревообрабатывающей промышленности, была начат грандиозный проект по созданию современной гибкой комплексной информационной системы. Проект получил название ALPINE. Его авторами являются доцент В. Пресняков и выпускник ЛТА (ныне системный администратор ОАО «Солнце») А. Ковешников. Разработка отдельных системных модулей проекта принадлежит доценту кафедры Н. Васильеву.

С 2003 года информационная система ALPINE внедрена и эксплуатируется на российских предприятиях финской компании UPM-Kymmene, в том числе на лесопильном заводе в г. Пестово Новгородской области. Завод по производству фанеры и тонкого шпона «ЮПМ-Кюммене Чудово» использует данную систему для учета всего производства (фанера, тонкий шпон, щепы, опилки и пр.), для учета реализации, а также для связи с другими программными продуктами, интегрированными в общую инфраструктуру компании.

Автоматизированная информационная система ALPINE разрабатывалась специально для комплексной работы предприятия по ведению и размещению заказов на производстве, контролю их выполнения, учету производства и реализации готовой продукции. Необходимость в использовании такой системы возникает из-за того, что другие доступные системы, как правило, не рассчитаны на обработку большого количества специфической информации и не имеют в своих стандартных поставках модулей для работы с производственными задачами и задачами по планированию производства. Именно наличие таких модулей, которые при необходимости позволяют осуществлять связь с производственными линиями, полностью отслеживают процесс производства и реализации и делают систему ALPINE уникальным средством, в котором в том числе учтена специфика лесной отрасли и деревообрабатывающих предприятий.

Особенно важным является свойство одновременной работы с данными большого числа работников предприятия. В основу разработки был положен принцип модульности, который позволяет разрабатывать и подключать модули, предназначенные для решения новых проблем и задач, возникающих на производстве.

Еще одной особенностью является наличие механизма, позволяющего осуществлять удаленное администрирование системы. Это в свою очередь дает возможность вносить изменения в конфигурацию системы, как в целом, так и в индивидуальные настройки пользователей системы, не останавливая работу системы.

Ядро системы ALPINE включает в себя набор программных модулей, отвечающих за следующий набор задач:

- администрирование пользователей и управления правами доступа,
- формирование представлений и управления интерфейсом,
- справочники системы,
- управление заказами,
- работа с множеством складов,
- реализации продукции,
- связь с внешними программными средствами.

По своей сути ядро системы ALPINE – это программный продукт со стандартным набором задач, который легко расширяется и модифицируется под конкретные нужды заказчика.

ALPINE позволяет работать со следующими группами задач:

- оформление заказов в различных валютах платежа,
- планирование производства,
- учет произведенной продукции и полный контроль хода выполнения заказов,
- ведение множества складов готовой продукции,
- учет реализации продукции для российских и зарубежных заказчиков,
- ведение книги продаж,
- подготовка платежных и прочих документов,
- возможность интегрирования с другими информационными и бухгалтерскими системами, используемыми на предприятии,
- формирование документов для печати в требуемом формате,

- связь с производственными линиями и автоматическое занесение данных о производстве,
- автоматическое получение информации о заказах из внешних систем при необходимости.

Гибкая система безопасности позволяет администратору системы настроить для каждого пользователя индивидуальные права доступа к отдельным задачам, модулям, режимам.

Открытая архитектура позволяет дополнять ядро системы модулями, являющимися специфическими для конкретного предприятия. Структура базы данных и пользовательский интерфейс спроектированы с возможностью использования штрих-кодирования при планировании, производстве, хранении и отгрузке продукции для упрощения ввода данных и отслеживания истории прохождения всей цепочки от планирования до реализации продукции.

ALPINE работает под управлением операционных систем семейства Microsoft Windows. Работу с данными ALPINE осуществляет в многопользовательском режиме. В качестве среды хранения и обработки данных используется SQL Server. Предусмотрена возможность доступа к данным из удаленных офисов в терминальном режиме.

Имеется возможность работы на нескольких заводах и с множеством складов, территориально удаленных друг от друга. Гибкое подключение новых складов и подразделений в рамках корпорации. Анализ и обработки информации разрешен в любой момент времени на любом рабочем месте с учетом предоставленного уровня доступа.

Уровень доступа к модулям, задачам, режимам, документам, а также права на просмотр, изменения и удаления данных и документов для каждого пользователя определяется администратором системы независимо от нахождения рабочего места пользователя. Для того чтобы можно было осуществлять удаленный доступ к информации, имеется полная поддержка работы в режиме терминал-сервер.

При разработке системы изначально включена многоязыковая поддержка пользовательского интерфейса, устанавливаемая для каждого пользователя индивидуально. Интерфейс и внутренняя структура базы данных рассчитаны на открытость для интеграции с другими АИС (например, Platinum, Scala, Wisa, SAP/R3 и т.д.).



Собственная система построения произвольных запросов к базе данных и экспорта результатов запросов в нужном формате расширяет возможности администрирования, получения необходимых результатов. Модуль экспорта и импорта данных для всех или части таблиц является прекрасным дополнением к стандартным способам резервирования и восстановления данных.

Основные технологические задачи по предметным направлениям, реализуемые в АИС ALPINE следующие:

### **Модуль работы с заказами**

Оформление заказов на производство (отгрузку) продукции в различных валютах платежа.

Оформление внутренних заказов для собственных нужд корпорации.

Использование любого количества прейскурантов для разных валют.

Полный контроль за ходом выполнения заказа, как в целом, так и по каждой позиции, включая показатели по плану, выполнению и отгрузке.

Формирование плана производства и отгрузки

Расчет свободной мощности производства.

### **Модуль работы со складами**

Ведение множества складов готовой продукции

Ведение картотеки продукции на складах и полный контроль движения продукции на складах по каждому товару.

Учет хранения продукции в нескольких единицах измерения.

Приемка на склад готовой продукции с производства, а также от любых подразделений корпорации.

Возврат продукции со склада на производство.

Осуществление внутреннего перемещения продукции между складами и подразделениями

Распределение продукции, поступившей на склады по заказам.

Перенос размещенной продукции с одного заказа на другой.

Полный контроль текущего состояния складов.

Проведение инвентаризации складов.

Учет выпуска продукции по сменам.

Отгрузка продукции со складов как по заказам, так и для внутренних нужд корпорации.

Формирование оборотных ведомостей за любой отчетный период.

### **Модуль реализации продукции**

Учет реализации продукции для российских и зарубежных заказчиков.

Ведение книги продаж.

Подготовка платежных и прочих документов, как для российских, так и для зарубежных покупателей.

Интегрирование с другими информационными и бухгалтерскими системами, используемыми на предприятии.

Учет оплаты и картотека расчетов с покупателями.

Расчет средней цены отгруженной продукции в любой указанной валюте в целом, по отдельным странам, по каждому покупателю.

В заключение представим технические характеристики текущей версии программного комплекса (по состоянию на 2008 год):

ALPINE на рабочих станциях работает под управлением операционной системы Microsoft Windows XP. В качестве среды хранения и обработки данных используется сервер баз данных Microsoft SQL Server 2000.

Рекомендуемые требования к серверу:

— Intel Xeon 2,4 GHz (или более мощный) на 2х процессорах (или больше):

- 2 GB RAM (или больше);
- RAID Controller с поддержкой RAID 5 и Ultra 320 SCSI channels;
- 20 GB дисковой памяти (или больше);
- Gigabit Ethernet server adapter;
- Операционная система – Windows 2003 Server.

При необходимости организации доступа к данным из удаленных офисов в терминальном режиме требуется установка терминал-сервера и соответствующих клиентов на рабочих станциях.

## ГРУППОВОЕ МНОГОКРИТЕРИАЛЬНОЕ ПРИНЯТИЕ РЕШЕНИЙ НА ОСНОВЕ АЛГЕБРЫ ПРЕДПОЧТЕНИЙ

**Введение.** Большинство логистических задач – это задачи многокритериального принятия решения (МПР) с большим количеством альтернатив и критериев. Сложность многокритериального подхода заключается в разнонаправленности критериев, в их смешанном характере (количественные и качественные критерии, причем последние имеют тенденцию доминировать). В настоящее время разработано большое количество подходов для решения подобных многокритериальных задач (AUT, АНР, ELECTRE), среди которых наиболее популярным является метод анализа иерархий (АНР/МАИ) [1].

Принятие решений в группе существенно отличается от индивидуального принятия решения. У каждого из членов группы имеется свой взгляд на проблему, и в этом случае возникает задача, как объединить суждения нескольких экспертов, каждый из которых имеет отличное от других мнение, отличную степень надежности или доверия к его оценкам.

Анализ существующих методов объединения экспертных суждений в задачах МПР показывает, что в разработанных методах решение производится либо на основе информации, полученной от одного источника, либо путем дополнительных преобразований типа усреднения, взвешивания, согласования различных оценок, которые могут привести к необоснованным и необратимым искажениям исходных данных. Поэтому необходимы такие математические методы, которые позволяли бы одновременно учитывать оценки, в том числе и противоречивые, всех экспертов без поиска компромисса между мнениями отдельных экспертов при условии, что не существует «главного» эксперта, и мнения всех экспертов считаются одинаково важными.

В связи с этим в данной модификации МАИ предлагается следующий подход к коллективному принятию решения.

1. Представление проблемы в виде иерархии посредством динамического обсуждения ее группой экспертов (аналогично МАИ).

2. Сбор информации посредством процедуры опроса экспертов в виде *M*-этапной схемы.

3. Математическая обработка и анализ собранной информации.

**Описание метода.** Рассмотрим суть предлагаемого подхода на примере задачи выбора поставщика. Пусть в результате обсуждения проблемы группой из 20 экспертов были выделены три критерия  $C_1$  – цена товара,  $C_2$  – надежность соблюдения сроков поставки,  $C_3$  – территориальная удаленность поставщика от потребителя, и четыре альтернативы  $A_1, A_2, A_3, A_4$  – поставщик №1, №2, №3, №4 соответственно, т.е. проблема представлена в виде двухуровневой иерархии.

Элементы любого уровня иерархии сравниваются друг с другом относительно их воздействия на элемент иерархии, расположенный на один уровень выше. Опрос экспертов в *M*-уровневой иерархии ведется в виде *M*-этапной схемы, и представляет собой одну из возможных формализованных процедур, сужающих множество Парето-оптимальных исходов.

Необходимо отметить, что зачастую при выполнении парных сравнений элементов (критериев или альтернатив) эксперт испытывает затруднение при сравнении их между собой, приписывая им числа в соответствии с выбранной шкалой, поскольку различие между отдельными элементами бывает столь незначительно, что его нельзя оценить. Чтобы устранить эту проблему в данной модификации предлагается рассматривать не только отдельные элементы, но и их группы (подмножества). Кроме того, в данном подходе предлагается облегчить процедуру парного сравнения элементов (групп элементов) между собой, разрешив эксперту высказывать только те предпочтения, которые он может предоставить.

Пусть  $A = \{A_1, A_2, \dots, A_n\}$  – множество альтернатив,  $Po(A)$  – множество мощности альтернатив, элементами которого являются всевозможные сочетания альтернатив  $B_i \subseteq A$ ,  $C = \{C_1, \dots, C_r\}$  – множество критериев,  $Po(C)$  – множество мощности критериев.

На первом этапе схемы опроса эксперт выделяет подмножество  $D_k \in Po(C)$  критериев из множества  $Po(C)$  и сравнивает его с другим подмножеством  $D_i \in Po(C)$  критериев, т.е. экспертом устанавливается предпочтение  $D_k \succ D_i$ . Для краткости будем обозначать  $D_{ki}$  предпочтение  $D_k \succ D_i$ .

Предполагается, что эксперты только сравнивают подмножества критериев, но не дают оценку степени сравнения. При этом если от  $n$ -го эксперта получена оценка  $D_k \succ D_i$ , то в клетку таблицы на пересечении  $k$ -й строки и  $i$ -го столбца ставится единица.

Следует заметить, что не все клетки в таблице подлежат заполнению. Предпочтения, в которых критерий (подмножество критериев)  $\tilde{D}_{ki} = \{C_i, C_j, \dots, C_l\}$  входит в обе части предпочтения  $D_{12}$ , т.е.  $D_k \cap D_i = \tilde{D}_{ki}$ , являются некорректными и требуют перевода в эквивалентные им предпочтения путем исключения из исходного предпочтения подмножества  $\tilde{D}_{ki} = \{C_i, C_j, \dots, C_l\}$ . После того как все эксперты выскажут свои предпочтения, подсчитывается общее число оценок  $c_{ki}$  по каждому предпочтению  $D_{ki}$ .

Пусть экспертами на первом этапе предоставлены следующие оценки:  $C_1 \succ C_2 - 2$  оценки,  $C_1 \succ C_3 - 4$ ,  $C_1 \succ C_2 C_3 - 3$ ,  $C_3 \succ C_1 - 1$ ,  $C_3 \succ C_2 - 2$ ,  $C_1 C_2 \succ C_3 - 4$ ,  $C_1 C_3 \succ C_2 - 1$ ,  $C_2 C_3 \succ C_1 - 1$ .

На втором этапе эксперт в соответствии с заданным критерием  $C_j$ ,  $C_j \in C$  высказывает предпочтение  $B_k \succ B_i$ , которое в дальнейшем будем обозначать  $V_{ki}$ . Если от  $n$ -го эксперта по критерию  $C_j$  получена оценка  $B_k \succ B_i$ , то в клетку таблицы на пересечении  $k$ -й строки и  $i$ -го столбца ставится единица. Процедура повторяется для всех критериев, после чего подсчитывается общее количество предпочтений  $a_{ki}^j$  высказанное всеми экспертами [4].

Ниже приведем полученные от экспертов оценки в соответствии с заданным критерием. По критерию  $C_1$ :  $A_1 \succ A_2 - 1$  оценка,  $A_1 \succ A_4 - 1$ ,  $A_1 \succ A_2 A_3 - 1$ ,  $A_1 \succ A_2 A_3 A_4 - 3$ ,  $A_2 \succ A_1 A_3 A_4 - 3$ ,  $A_3 \succ A_1 - 1$ ,  $A_3 \succ A_4 - 2$ ,  $A_3 \succ A_1 A_4 - 2$ ,  $A_3 \succ A_1 A_2 A_4 - 2$ ,  $A_1 A_2 \succ A_3 A_4 - 2$ ,  $A_2 A_3 \succ A_4 - 1$ . По критерию  $C_2$ :  $A_1 \succ A_3 - 2$ ,  $A_1 \succ A_3 A_4 - 1$ ,  $A_1 \succ A_2 A_3 A_4 - 1$ ,  $A_2 \succ A_1 A_3 A_4 - 3$ ,  $A_3 \succ A_1 A_4 - 1$ ,  $A_4 \succ A_2 - 1$ ,  $A_4 \succ A_1 A_2 A_3 - 1$ ,  $A_1 A_2 \succ A_3 A_4 - 3$ ,  $A_1 A_3 \succ A_2 A_4 - 1$ ,  $A_2 A_3 \succ A_1 A_4 - 2$ ,  $A_2 A_4 \succ A_3 - 3$ ,  $A_1 A_2 A_4 \succ A_3 - 1$ . По критерию  $C_3$ :  $A_1 \succ A_2 A_3 A_4 - 1$ ,  $A_3 \succ A_1 - 1$ ,  $A_3 \succ A_1 A_2 - 5$ ,  $A_3 \succ A_1 A_2 A_4 - 1$ ,  $A_1 A_2 \succ A_3 - 1$ ,  $A_1 A_3 \succ A_2 A_4 - 4$ ,  $A_2 A_3 \succ A_1 A_4 - 2$ ,  $A_3 A_4 \succ A_1 - 1$ ,  $A_3 A_4 \succ A_1 A_2 - 2$ ,  $A_1 A_2 A_4 \succ A_3 - 1$ ,  $A_2 A_3 A_4 \succ A_1 - 1$ .

## Анализ собранной информации и алгебра предпочтений

Выделим четыре типа предпочтений  $D_k \succ D_i$ : 1) левая и правая части предпочтения содержат по одному элементу, т. е.  $D_k \in C, D_i \in C, C_l \succ C_m, l \neq m$  (например  $C_1 \succ C_2$ ); 2) левая часть предпочтения содержит один элемент, а правая – несколько, т.е.  $D_k \in C, D_i \subseteq C, D_k \succ D_i = C_l \succ C_j C_m$  (например  $C_1 \succ C_2 C_3$ ); 3) левая часть предпочтения содержит несколько элементов, а правая – один, т.е.  $D_k \subseteq C, D_i \in C, D_k \succ D_i = C_1 \dots C_j \succ C_m$  (например  $C_1 C_2 \succ C_3$ ); 4) левая и правая части предпочтений составлены из групп элементов, т.е.  $D_k \subseteq C, D_i \subseteq C, D_k \succ D_i = C_1 \dots C_j \succ C_m \dots C_n$  (например  $C_1 C_2 \succ C_3 C_4$ ).

Первый тип предпочтений будем называть элементарными или *базовыми предпочтениями*. Элементарное предпочтение есть не что иное, как парное качественное сравнение двух элементов (критериев)  $C_i$  и  $C_k$ . Можно записать множество «элементарных» предпочтений критериев как:

$$\Theta = \{\{C_i\} \succ \{C_k\}, \forall i, k \in \{1, 2, \dots, r\}, i \neq k\} = \{\theta_1, \theta_2, \dots, \theta_\tau\}, \tau = r \times (r - 1).$$

Предпочтение второго типа (составное предпочтение)  $C_1 \succ C_2 C_3$  означает, что критерий  $C_1$  одновременно предпочтительнее и критерия  $C_2$ , и критерия  $C_3$ . Его можно представить как совокупность двух пересекающихся элементарных предпочтений  $C_1 \succ C_2 C_3 \equiv \{C_1 \succ C_2\} \wedge \{C_1 \succ C_3\}$ . Здесь и далее символ  $\wedge$  означает логическое «и», а символ  $\equiv$  означает эквивалентные предпочтения.

Предпочтение третьего типа (неточное предпочтение)  $C_1 C_2 \succ C_3$  означает, что либо критерий  $C_1$ , либо критерий  $C_2$  предпочтительнее критерия  $C_3$ , т.е. оно представляет собой объединение двух элементарных предпочтений  $C_1 C_2 \succ C_3 \equiv (C_1 \succ C_3) \vee (C_2 \succ C_3)$ , где символ  $\vee$  означает логическое «или».

Таким образом, любое предпочтение вида  $\{C_i, C_j\} \succ \{C_k, C_m\}$  посредством выполнения над ним логических операций можно представить как эквивалентную совокупность элементарных предпочтений следующим образом:  $\{C_i C_j\} \succ \{C_k C_l\} \equiv (\{C_i \succ C_k\} \vee \{C_j \succ C_k\}) \wedge (\{C_i \succ C_l\} \vee \{C_j \succ C_l\})$ .

## Определение весов критериев и альтернатив

Определение весов (вероятностей) элементарных предпочтений посредством теории вероятностей не представляется возможным. Наиболее подходящим математическим аппаратом является теория свидетельств или теория Демпстера–Шейфера [2, 3].

Тогда множество элементарных предпочтений можно рассматривать как множество элементарных исходов (универсальное множество), а все высказанные предпочтения  $D_{ki}$  – как подмножества универсального множества. В соответствии с теорией свидетельств, для каждой клетки итоговой расширенной матрицы (фокального элемента) определим базовую вероятность

$$m(D_k \succ D_i) = m(D_{ki}) = \frac{c_{ki}}{N}, \text{ здесь } N - \text{ число оценок.}$$

$$\text{Например, } m(D_1 \succ D_2) = m(D_{12}) = \frac{2}{20} = 0,1.$$

Далее используя базовые вероятности предпочтений, для каждого фокального элемента  $D_{ki}$  необходимо определить нижнюю и верхнюю границы его вероятности, т.е. функции доверия  $Bel(D_{ki})$  и правдоподобия  $Pl(D_{ki})$ . В связи с тем, что алгебра элементарных исходов заменена нами алгеброй элементарных предпочтений, вычисление функций доверия и правдоподобия будет несколько отличаться от стандартной обработки, используемой в теории свидетельств.

*Утверждение 1. Предположим, что имеется информация в виде базовых вероятностей предпочтений вида*

$$D_{ki} = D_k \succ D_i, D_k \subset C, D_i \subset C, D_k \cap D_i = \emptyset,$$

$$D_{lm} = D_l \succ D_m, D_l \subset C, D_m \subset C, D_l \cap D_m = \emptyset.$$

*Тогда функция доверия предпочтения  $D_{ki}$  равна*

$$Bel(D_{ki}) = \sum_{D_l: D_l \subset D_k} \sum_{D_m: D_l \subset D_m, D_l \cap D_m = \emptyset} m(D_{lm}). \quad (1)$$

*Функция правдоподобия определяется через функцию доверия как*

$$Pl(D_{ki}) = 1 - Bel(D_{ik}). \quad (2)$$

### Доказательство

Предпочтение  $D_{ki}$  можно представить как объединение множеств предпочтений вида  $C_j \succ D_i, C_j \in D_k$ . Следовательно, каждое такое предпочтение входит в  $D_{ki}$ , т.е.  $(C_j \succ D_i) \subseteq (D_k \succ D_i)$ . Поэтому базовые вероятности этих предпочтений участвуют в вычислении функции доверия  $Bel(D_{ki})$ .

Заметим, что если выполняется условие  $D_l \subseteq D_k$ , то любое предпочтение  $D_l \succ D_i$  можно представить как объединение части предпочтений  $C_j \succ D_i, C_j \in D_k$ . Следовательно, выполняется условие  $(D_l \succ D_i) \subseteq (D_k \succ D_i)$ , и базовые вероятности любых предпочтений вида  $D_l \succ D_i$  при  $D_l \subseteq D_k$  участвуют в вычислении функции доверия  $Bel(D_{ki})$ , т.е.

$$Bel(D_{ki}) \leq \sum_{D_l: D_l \subseteq D_k} m(D_l \succ D_i).$$

Рассмотрим подмножество критериев  $D_m = (Q, D_i)$ , где  $Q \cap D_i = \emptyset$ . Заметим, что в этом случае  $D_m \supseteq D_i$  или  $D_i \subseteq D_m$ . Тогда условие  $(D_l \succ D_m) \subseteq (D_l \succ D_i)$  следует из

$$D_l \succ D_m = D_l \succ (Q, D_i) = (D_l \succ Q) \cap (D_l \succ D_i) \subseteq (D_l \succ D_i).$$

Таким образом, если  $(D_l \succ D_m) \subseteq (D_l \succ D_i)$ , а  $(D_l \succ D_i) \subseteq (D_k \succ D_i)$ , то  $(D_l \succ D_m) \subseteq (D_k \succ D_i)$  и базовые вероятности любых предпочтений вида  $(D_l \succ D_m)$  при  $D_i \subseteq D_m$  участвуют в вычислении функции доверия  $Bel(D_{ki})$ .

Суммируя все учтенные вероятности, получаем (1). Выражение (2) для функции правдоподобия непосредственно следует из свойств функций доверия и правдоподобия.  $\square$

В качестве примера определим функцию доверия предпочтения  $D_{61} = D_6 \succ D_1 = C_2 C_3 \succ C_1$ . В терминах утверждения 1 запишем

$$D_k = \{C_2, C_3\}, D_i = \{C_1\}, D_k \cap D_i = \emptyset.$$



Рассмотрим сначала всевозможные  $D_l$  такие, что  $D_l \subseteq D_k$ :  
 $D_l = \{C_2\}, \{C_2\} \subseteq \{C_2, C_3\}, \quad D_l = \{C_3\}, \{C_3\} \subseteq \{C_2, C_3\}, \quad D_l = \{C_2 C_3\}, \{C_2, C_3\} \subseteq \{C_2, C_3\}$ .  
Теперь для каждого  $D_l$  определим всевозможные  $D_m$  при условиях  $D_i \subseteq D_m$   
и  $D_l \cap D_m = \emptyset$ :

$$\begin{aligned} D_l = \{C_2\}, \{C_1\} \subseteq \{C_1\}, \{C_2\} \cap \{C_1\} &= \emptyset \\ D_l = \{C_2\}, \{C_1\} \subseteq \{C_1, C_3\}, \{C_2\} \cap \{C_1, C_3\} &= \emptyset \\ D_l = \{C_3\}, \{C_1\} \subseteq \{C_1\}, \{C_3\} \cap \{C_1\} &= \emptyset \\ D_l = \{C_3\}, \{C_1\} \subseteq \{C_1, C_2\}, \{C_3\} \cap \{C_1, C_2\} &= \emptyset \\ D_l = \{C_2, C_3\}, \{C_1\} \subseteq \{C_1\}, \{C_2, C_3\} \cap \{C_1\} &= \emptyset \end{aligned}$$

Таким образом

$$\begin{aligned} Bel(D_{61}) &= Bel(C_2 C_3 \succ C_1) = m(C_2 \succ C_1) + \\ &+ m(C_2 \succ C_1 C_3) + m(C_3 \succ C_1) + m(C_3 \succ C_1 C_2) + m(C_2 C_3 \succ C_1) = \\ &= m(\theta_3) + m(\theta_3 \wedge \theta_4) + m(\theta_5) + m(\theta_5 \wedge \theta_6) + m(\theta_3 \vee \theta_5) = \\ &= 0 + 0 + 0,05 + 0 + 0,15 = 0,2 \end{aligned}$$

Зная функцию доверия предпочтения  $D_{61} = C_2 C_3 \succ C_1$ , определим функцию правдоподобия предпочтения  $D_{16} = C_1 \succ C_2 C_3$  следующим образом:

$$Pl(D_{16}) = 1 - Bel(D_{61}) = 1 - 0,20 = 0,80.$$

В то же время следует отметить, что функции доверия и правдоподобия высказанных предпочтений сами по себе не несут информации о значимости (вероятности) критериев. Здесь возникает вопрос: как от функций доверия и правдоподобия предпочтений критериев перейти к вероятностям критериев?

Одним из возможных подходов является рассмотрение в качестве вероятности критерия  $p(C_j)$  вероятности предпочтения  $C_j \succ C$ . Располагая неточной информацией о предпочтениях, мы не можем определить точные значения весов (вероятностей) критериев, а способны только указать область их определения, которая задается следующей системой неравенств:

$$\left\{ \begin{array}{l} Bel(D_k \succ C) \leq \sum_{j: C_j \in D_k} p_j \leq Pl(D_k \succ C) \\ \sum_{j=1}^r p_j = 1 \\ p_j \geq 0, \forall j, j = 1, \dots, r \end{array} \right. \quad (3)$$

Здесь  $p_j$  можно рассматривать как вес (вероятность) критерия  $C_j$ . Система неравенств образована только линейными неравенствами. Поэтому область определения критериев является выпуклой, и существует некоторое множество  $P$  возможных распределений вероятностей  $\mathbf{p} = (p_1, \dots, p_r)$ , удовлетворяющее всем неравенствам (3) одновременно.

Аналогичные рассуждения можно привести и для альтернатив. Для каждого фокального элемента расширенной итоговой матрицы парных сравнений предпочтений альтернатив определим базовую вероятность

$$m(B_{ki} | C_j) = \frac{a_{ki}^j}{N_A^j}.$$

Для определения функций доверия и правдоподобия перепишем Утверждение 1 в обозначениях предпочтений альтернатив. Предположим, что имеется информация в виде базовых вероятностей  $m(B_{ki} | C_j)$  и  $m(B_{lm} | C_j)$  предпочтений альтернатив вида  $B_{ki} = B_k \succ B_i$ ,  $B_k \subset \mathbf{A}$ ,  $B_i \subset \mathbf{A}$ ,  $B_k \cap B_i = \emptyset$ ,  $B_{lm} = B_l \succ B_m$ ,  $B_l \subset \mathbf{A}$ ,  $B_m \subset \mathbf{A}$ ,  $B_l \cap B_m = \emptyset$ .

Тогда функция доверия предпочтения  $B_{ki}$  в зависимости от критерия  $C_j$  равна

$$Bel(B_{ki} | C_j) = \sum_{B_l: B_l \subseteq B_k} \sum_{B_m: B_l \subseteq B_m, B_l \cap B_m = \emptyset} m(B_{lm} | C_j). \quad (4)$$

Функция правдоподобия определяется через функцию доверия как

$$Pl(B_{ki} | C_j) = 1 - Bel(B_{ik} | C_j). \quad (5)$$

### Формирование глобального критерия

Метод для комбинирования и обработки неполной информации полностью зависит от критерия принятия решений. Один из наиболее применяемых критериев – линейная функция  $F$

$$F(\mathbf{p}, \mathbf{u}_k) = \sum_{j=1}^r (p_j \cdot u_{jk}),$$

где  $\mathbf{p}=(p_1,\dots,p_r)$ – вектор весов или значимостей критериев,  $\mathbf{u}_k=(u_{1k},\dots,u_{rk})$ ,  $k=1,\dots,n$  – вектор весов или полезностей  $k$ -й альтернативы в соответствии с критерием из множества  $\mathbf{C}=\{C_1,\dots,C_r\}$ .

Очевидно, что при наличии только ограниченной информации о векторах  $\mathbf{p}$  и  $\mathbf{u}_k$ , можно получить только ограниченную информацию о функции  $F$ , т.е. функция  $F$  принадлежит замкнутому интервалу. Тогда выбор наилучшей альтернативы может быть основан на сравнении интервалов функции  $F$ . Существует большое количество методов сравнения. Здесь предлагается использовать метод, основанный на параметре осторожности или на параметре пессимизма  $\gamma \in [0,1]$ . Наилучшая альтернатива из всего множества возможных альтернатив выбирается так, что линейная комбинация  $Y = \gamma \cdot \inf F + (1 - \gamma) \cdot \sup F$  достигает своего максимума.

В соответствии со схемой опроса экспертов на втором этапе базовые вероятности предпочтений альтернатив являются условными в зависимости от критерия  $C_j$ . Зафиксируем одно распределение  $\mathbf{p}=(p_1,\dots,p_r)$  из  $\mathbf{P}$ . Тогда, используя линейную свертку, для каждого подмножества альтернатив мы можем записать комбинированную базовую вероятность следующим образом:

$$F = m_p(B_{ki}) = \sum_{j=1}^r m(B_{ki} | C_j) \cdot p_j, \quad p \in \mathbf{P},$$

$$\inf F = \inf_{p \in \mathbf{P}} Bel_p(B_{ki}), \quad \sup F = \sup_{p \in \mathbf{P}} Pl_p(B_{ki}),$$

$$Bel_p(B_{ki}) = \sum_{j=1}^r p_j \cdot \sum_{B_l: B_l \subseteq B_k} \sum_{B_m: B_l \subseteq B_m, B_l \cap B_m = \emptyset} m(B_{lm} | C_j), \quad Pl_p(B_{ki}) = 1 - Bel_p(B_{ik}).$$

Заметим, что полученные функции доверия и правдоподобия зависят от  $p$  линейно. Следовательно, можно вычислить нижнюю и верхнюю границы функций доверия и правдоподобия на основе задач линейного программирования при ограничениях (3).

В качестве примера рассмотрим определение функции доверия для альтернативы  $A_i$ . Система ограничений будет иметь вид:

$$\begin{cases} 0,15 \leq p_1 \leq 0,8 \\ 0 \leq p_2 \leq 0,6 \\ 0 \leq p_3 \leq 0,45 \\ 0,55 \leq p_1 + p_2 \leq 1 \\ 0,4 \leq p_1 + p_3 \leq 1 \\ 0,2 \leq p_2 + p_3 \leq 0,85 \\ p_1 + p_2 + p_3 = 1 \end{cases}$$

$$Bel(A_1 \succ \mathbf{A}) = p_1 \cdot Bel(A_1 \succ \mathbf{A} | C_1) + p_2 \cdot Bel(A_1 \succ \mathbf{A} | C_2) + p_3 \cdot Bel(A_1 \succ \mathbf{A} | C_3) \rightarrow \min$$

$$Bel(A_1 \succ \mathbf{A}) = p_1 \cdot 0,15 + p_2 \cdot 0,05 + p_3 \cdot 0,05 \rightarrow \min .$$

Решая задачу линейного программирования на минимум, находим

$$Bel(A_1 \succ \mathbf{A}) = 0,15 \cdot 0,15 + 0,6 \cdot 0,05 + 0,25 \cdot 0,05 = 0,065 .$$

Результат сведем в таблицу:

	$A_1$	$A_2$	$A_3$	$A_4$
$Bel(B_i)$	0,0650	0,0825	0,0275	0,0000
$Pl(B_i)$	0,6300	0,7400	0,685	0,4525

Если  $\gamma = 1$ , то мы сравниваем только нижние границы для  $F$  по всем альтернативам, и принимаем пессимистическое решение, при котором альтернативы могут быть ранжированы следующим образом  $A_2 \succ A_1 \succ A_3 \succ A_4$ . Если  $\gamma = 0$ , то сравниваются только верхние границы, и мы принимаем оптимистическое решение, и  $A_2 \succ A_3 \succ A_1 \succ A_4$ . При  $\gamma = 0,59$  альтернативы  $A_1$  и  $A_3$  несравнимы (точка неустойчивого решения).

### Библиографический список

1. Саати Т. Принятие решений. Метод анализа иерархий. Радио и связь. М.: 1993.
2. Dempster A. P. Upper and lower probabilities induced by a multi-valued mapping. *Annales of Mathematical Statistics*, 38: 325–339, 1967.
3. Shafer G. *A Mathematical Theory of Evidence*. Princeton University Press, 1976.
4. Уткин Л. В., Симанова Н. В. Обобщение метода анализа иерархий для принятия решений при неточных сравнениях с использованием теории Демпстера–Шейфера / «Известия ОрелГТУ», №4/268(535), 2007. С. 223–229.

**С. П. Хабаров**, кандидат технических наук, доцент,  
**Е. Л. Колмогорцев**, кандидат технических наук, доцент

## **К ВОПРОСУ ОБ ОЦЕНКЕ ВРЕМЕНИ РЕАКЦИИ РАСПРЕДЕЛЕННОЙ СИСТЕМЫ УПРАВЛЕНИЯ С РЕЗЕРВИРОВАНИЕМ СЕТЕВОЙ КОММУНИКАЦИОННОЙ ПОДСИСТЕМЫ**

**Введение.** При построении систем управления резервирование коммуникационной среды позволяет увеличить как надежность системы – при работе в режиме переключения резерва, так и надежность и производительность одновременно – при работе в режиме балансировки нагрузки [1].

На время реакции любой системы управления налагается некоторое верхнее ограничение  $t_{\text{реакции}} \leq T^*$ , которое является критерием работоспособности системы с точки зрения ее производительности.

В системах управления, работающих в режиме балансировки нагрузки, при отказах коммуникационного оборудования (коммутаторов) производительность системы снижается постепенно (деградация) [1]. При накоплении определенного числа отказов производительность системы снижается на столько, что уже не обеспечивает требуемых ограничений на время реакции.

В статье рассматриваются иерархические распределенные системы управления, в которых сбор и первичная обработка информации производится узлами нижнего уровня иерархии – узлами ввода/вывода (УВВ), а управление обработкой – узлами верхнего уровня – управляющими компьютерами (УК). Обмен данными между узлами разных уровней иерархии осуществляется посредством резервированной коммуникационной среды, построенной на коммутаторах, по протоколу «запрос – ответ» (рис. 1).

**Постановка задачи.** Проектируемая система управления должна обладать производительностью, обеспечивающей необходимое время реакции (т.е. производительность, при которой выполняется заданное ограничение на время реакции). Таким образом, при проектировании системы управления необходимо для каждого рассматриваемого варианта построения системы, различающихся количеством вычислительных узлов в систе-

ме, составом коммуникационного оборудования, характеристиками производительности оборудования, оценить время реакции системы.

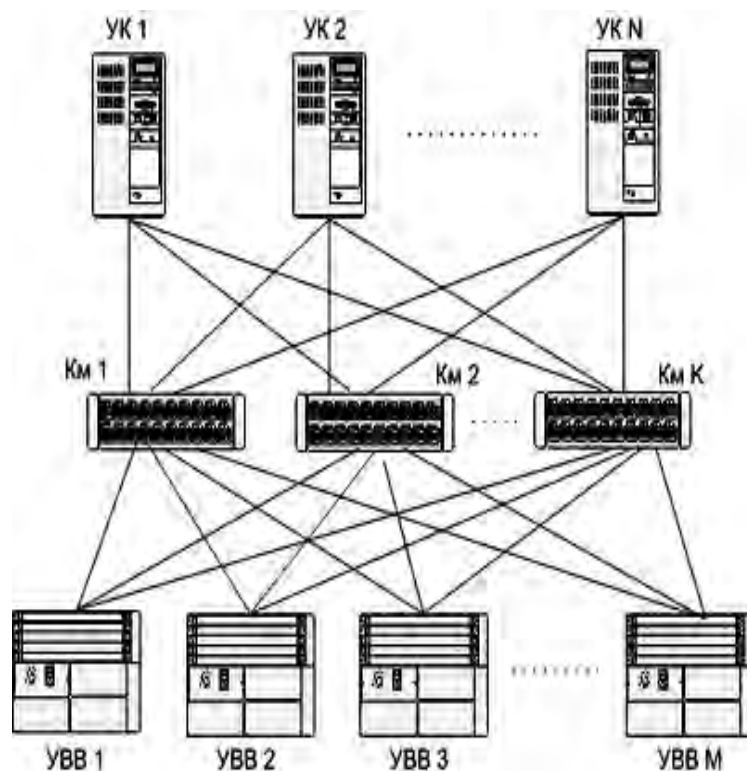


Рис. 1. Система управления

**Разработка модели.** Для решения поставленной задачи построим аналитическую модель рассмотренного выше класса систем [2]. Будем считать, что заданы следующие параметры системы:

- $M$  – количество УВВ;
- $N$  – количество управляющих компьютеров (УК);
- $K$  – количество коммутаторов;
- $T^*$  – ограничение на время реакции системы управления.

Пусть также известны параметры производительности оборудования. Рассматриваемая система может быть представлена (рис. 2) в виде сетевой системы массового обслуживания (СеМО).

Управляющие компьютеры  $УК_1 \dots УК_N$  генерируют запросы к УВВ длиной  $l_1$  с интенсивностями  $\lambda_1 \dots \lambda_N$ . Запросы через коммутаторы  $Км_1 \dots Км_K$  поступают в  $УВВ_1 \dots УВВ_M$ . После обработки запросов УВВ формируют ответы длиной  $l_2$ , поступают в УК и обрабатываются там.

Таким образом, временем реакции системы можно считать суммарное время пребывания в системе пакета-запроса и пакета-ответа.

Для расчета выше представленной СеМО, необходимо определить интенсивности заявок, поступающих на вход каждого узла сети и законы распределения времени между соседними заявками, а также среднее время и распределение времени обработки заявок (пакетов-запросов и пакетов-ответов) в узлах сети.

Известно, что поток, являющийся суперпозицией нескольких потоков, сравнимых по своему влиянию на суммарный поток, можно рассматривать как простейший [3,5]. Также, из теории массового обслуживания известно, что выходящий поток из СМО с  $m$  устройствами с ожиданием при простейшем входящем потоке с параметром  $\lambda$  и экспоненциальном распределении времени обслуживания с параметром  $\mu$  есть простейший поток с параметром  $\lambda_{\text{ВЫХ}} = \min\{\lambda, \mu\}$  [4].

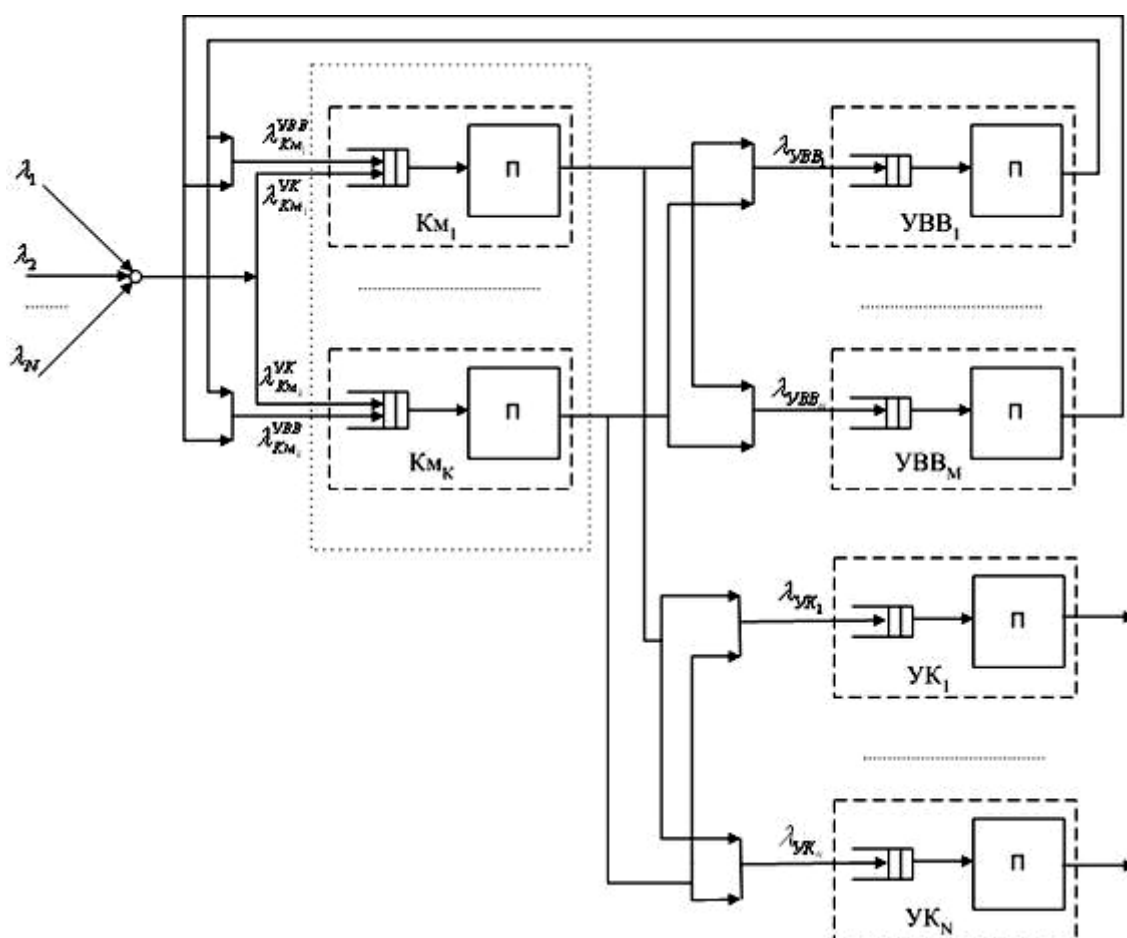


Рис. 2. Модель системы управления в виде СеМО

Для определения интенсивностей заявок на входе узлов сети примем следующее допущение: будем рассматривать только стационарный режим работы СеМО. Заявки из сети не удаляются, все узлы имеют бесконечные очереди. Для сети, работающей в установившемся режиме справедливы законы о суммарных потоках [4].

Наиболее вероятно, что поток запросов от УК<sub>n</sub> в общем случае является детерминированным либо потоком с ограниченным последствием (поток Пальма). Но т.к. на вход коммутатора КМ<sub>k</sub> поступает суммарный поток заявок (запросов) от УК<sub>1</sub>...УК<sub>n</sub>, то для упрощения построения аналитической модели примем допущение, что суммарный поток заявок-запросов на входе коммутатора КМ<sub>k</sub> имеет экспоненциальное распределение с интенсивностью  $\lambda_{КМ_k}$ .

Для определения интенсивности  $\lambda_{КМ_k}$  потока запросов, проходящих через коммутатор КМ<sub>k</sub>, введем в рассмотрение матрицу вероятностей  $P1 = \| p1_{nk} \|_{N \times K}$ , элемент которой  $p1_{nk}$  – характеризует вероятность передачи запросов от УК<sub>n</sub> через коммутатор КМ<sub>k</sub> ( $\sum_{k=1}^K p1_{nk} = 1$ , для  $n=1...N$ ). Тогда интенсивность потока заявок от всех УК на входе КМ<sub>k</sub> равна:

$$\lambda_{КМ_k}^{YK} = \sum_{n=1}^N \lambda_n p1_{nk} . \quad (1)$$

Также, на вход коммутатора КМ<sub>k</sub> поступает поток ответов от УВВ  $\lambda_{КМ_k}^{YBB}$ , который будет рассмотрен ниже, т.к. на данном этапе для определения его параметров нет достаточных исходных данных – не определены параметры входного потока запросов в УВВ и распределение времени обслуживания запросов в УВВ. При построении модели принято допущение, что времена обслуживания пакетов-запросов и пакетов-ответов в коммутаторе являются случайными и имеют, в общем случае, произвольное распределение [6,7].

Рассмотрим поток заявок-запросов на входе УВВ<sub>m</sub>. Т.к. на вход УВВ<sub>m</sub> поступает суммарный поток заявок с коммутаторов КМ<sub>1</sub>...КМ<sub>k</sub>, примем допущение, что поток заявок в УВВ<sub>m</sub> имеет экспоненциальное распределение. Для определения интенсивности потока заявок (запросов от УК) на входе УВВ<sub>m</sub> введем в рассмотрение матрицу вероятностей  $P2 = \| p2_{nm} \|_{N \times M}$ ,



элемент которой  $p2_{nm}$  – характеризует вероятность передачи запросов от УК<sub>n</sub> в УВВ<sub>m</sub> ( $\sum_{m=1}^M p2_{nm} = 1$ , для  $n=1 \dots N$ ). Тогда суммарная интенсивность потока заявок к УВВ<sub>m</sub> от всех УК равна:

$$\lambda_{УВВ_m} = \sum_{k=1}^K \sum_{n=1}^N \lambda_n p1_{nk} p2_{nm} . \quad (2)$$

Для упрощения построения аналитической модели будем считать, что время обслуживания заявок в УВВ и в УК распределено по экспоненциальному закону. Таким образом, УВВ<sub>m</sub> можно представить СМО вида М/М/1.

Т. к. мы приняли допущение о стационарном режиме работы системы и учитываем экспоненциальность входного и выходного потоков для УВВ<sub>1</sub>...УВВ<sub>M</sub>, то интенсивность потока ответов от УВВ, поступающих на вход коммутатора Км<sub>k</sub> равна интенсивности потока запросов, проходящих через этот коммутатор:

$$\lambda_{Км_k}^{УВВ} = \lambda_{Км_k}^{УК} , \quad (3)$$

имеет экспоненциальное распределение. Таким образом коммутатор Км<sub>k</sub> можно представить в виде СМО вида М/Г/1. Суммарный поток заявок-запросов и заявок-ответов, поступающий на вход коммутатора Км<sub>k</sub> имеет интенсивность

$$\lambda_{Км_k} = \lambda_{Км_k}^{УК} + \lambda_{Км_k}^{УВВ} = 2 \lambda_{Км_k}^{УК} . \quad (4)$$

Т.к. на вход УК<sub>n</sub> поступает суммарный поток ответов, проходящий через коммутаторы Км<sub>1</sub>...Км<sub>k</sub>, то примем допущение, что он также имеет экспоненциальное распределение. Интенсивность потока ответов, поступающих на обработку в УК<sub>n</sub>, равна интенсивности запросов, генерируемых данным управляющим компьютером:

$$\lambda_{УК_n} = \lambda_n . \quad (5)$$

Из выше сказанного следует, что управляющий компьютер УК<sub>n</sub> можно представить в виде СМО М/М/1.

Для расчета сети массового обслуживания необходимо также знать среднее время обслуживания заявок в каждом узле. Среднее время обслу-

живания определяется производительностью вычислительного узла, а также набором и сложностью решаемых узлом задач. При принятых допущениях (пакеты-запросы и пакеты-ответы от всех УК и УВВ имеют одинаковую длину, все однотипные узлы в системе одинаковы) имеем, что среднее время обслуживания в однотипных узлах одинаково. Обозначим:

$b_{req}$  – среднее время обслуживания в коммутаторе запроса от УК к УВВ;

$b_{ask}$  – среднее время обслуживания в коммутаторе ответа от УВВ к УК;

$b_{УВВ}$  – среднее время обслуживания в УВВ запроса от УК;

$b_{УК}$  – среднее время обслуживания запроса в УК ответа от УВВ.

Таким образом, зная  $\lambda_1 \dots \lambda_N$ ,  $P1$ ,  $P2$ ,  $b_{req}$ ,  $b_{ask}$ ,  $b_{УВВ}$  и  $b_{УК}$ , можно вычислить время пребывания заявки (пакета-запроса или пакета-ответа) в каждом узле системы, через которые проходит заявка.

Обозначим через  $T_{Км_k}$ ,  $T_{УВВ_m}$ ,  $T_{УК_n}$  – среднее время пребывания заявок, соответственно в  $Км_k$ ,  $УВВ_m$ ,  $УК_n$ .

Определим среднее время реакции системы как  $\bar{T}_{реакции} = \bar{T}_{Км} + \bar{T}_{УВВ} + \bar{T}_{УК}$ , где  $\bar{T}_{Км}$  – среднее по всем  $Км$  время пребывания заявки в коммутаторе;  $\bar{T}_{УВВ}$  – среднее по всем УВВ время пребывания заявки в устройстве ввода/вывода;  $\bar{T}_{УК}$  – среднее по всем УК время пребывания заявки в управляющем компьютере. Вычислим  $T_{Км}$ ,  $T_{УВВ}$ ,  $T_{УК}$  и определим время реакции системы.

### Коммутатор (М/Г/1):

$T_{Км_k} = b_{Км} + \frac{\rho_{Км_k} b_{Км} (1 + C_b^2)}{2(1 - \rho_{Км_k})}$ , где  $b_{Км}$  – среднее время обслуживания заявки

(пакета-запроса или пакета-ответа) в коммутаторе;  $C_b = \frac{\sigma_{Км}}{b_{Км}}$  – коэффициент

вариации,  $\rho_{Км_k} = \lambda_{Км_k} b_{Км}$  – загрузка (коэффициент использования) коммутатора.

Согласно (3) вероятности обслуживания в коммутаторе пакета-запроса и пакета-ответа одинаковы и равны  $p_{req} = p_{ask} = 0,5$ . Тогда,

$$b_{Км} = \frac{b_{req} + b_{ask}}{2},$$

$$\sigma_{KM}^2 = (b_{req} - b_{KM})^2 * p_{req} + (b_{ask} - b_{KM})^2 * p_{ask} = \frac{(b_{req} - b_{KM})^2 + (b_{ask} - b_{KM})^2}{2},$$

$$\bar{T}_{KM} = \sum_{k=1}^K T_{KM_k} * \frac{\lambda_{KM_k}}{\sum_{k=1}^K \lambda_{KM_k}} = \sum_{k=1}^K T_{KM_k} * p_{KM_k}, \text{ где } p_{KM_k} = \frac{\lambda_{KM_k}}{\sum_{k=1}^K \lambda_{KM_k}} \text{ есть вероятность то-}$$

го, что заявка обслуживается в  $KM_k$ .

### УВВ (М/М/1)

$$T_{УВВ_m} = \frac{b_{УВВ}}{1 - \rho_{УВВ_k}}, \text{ где } \rho_{УВВ_k} = \lambda_{УВВ_m} b_{УВВ} - \text{загрузка УВВ}_m.$$

$$\bar{T}_{УВВ} = \sum_{m=1}^M T_{УВВ_m} * \frac{\lambda_{УВВ_m}}{\sum_{m=1}^M \lambda_{УВВ_m}} = \sum_{m=1}^M T_{УВВ_m} * p_{УВВ_m}, \text{ где } p_{УВВ_m} = \frac{\lambda_{УВВ_m}}{\sum_{m=1}^M \lambda_{УВВ_m}} \text{ есть вероят-}$$

ность того, что заявка обслуживается в  $УВВ_m$ .

### УК (М/М/1)

$$T_{УК_n} = \frac{b_{УК}}{1 - \rho_{УК_n}}, \text{ где } \rho_{УК_n} = \lambda_{УК_n} b_{УК} - \text{загрузка УК}_n.$$

$$\bar{T}_{УК} = \sum_{n=1}^N T_{УК_n} * \frac{\lambda_{УК_n}}{\sum_{n=1}^N \lambda_{УК_n}} = \sum_{n=1}^N T_{УК_n} * p_{УК_n}, \text{ где } p_{УК_n} = \frac{\lambda_{УК_n}}{\sum_{n=1}^N \lambda_{УК_n}} \text{ есть вероятность то-}$$

го, что заявка обслуживается в  $УК_n$ .

Ввиду принятых допущений следует, что рассчитанные характеристики аналитической модели носят оценочный характер. Использование экспоненциальных распределений является грубым допущением и не обеспечивает высокой точности. Для получения более точных результатов необходимо применять имитационное моделирование.

Для моделирования СеМО была построена имитационная модель на языке GPSS.

Модель состоит из следующих блоков:

- 1) блок(и) генерации запросов от УК,
- 2) блок, имитирующий работу коммутаторов,
- 3) блок, имитирующий работу УВВ,
- 4) блок, имитирующий работу УК,
- 5) блок сбора статистики.

Также в состав модели входят функции, описывающие вероятность передачи запросов от  $УК_n$  в  $УВВ_m$  через  $К_{m_k}$ , и переменные, задающие распределение времен обработки заявок в узлах СеМО (в УК, УВВ и КМ).

Такая структура модели позволяет менять количество узлов сети, варьировать параметры потоков заявок и параметры обслуживания, изменяя только значения соответствующих переменных или функций (или добавляя новые). Все блоки модели при этом остаются неизменными. Исключение составляют блоки генерации запросов от УК – таких блоков должно быть по количеству УК в моделируемой системе.

Транзакты в модели имитируют пакеты-запросы и пакеты ответы. До обработки транзакта в блоке, моделирующем работу УВВ, транзакт имитирует пакет-запрос, после – пакет-ответ. Время пребывания транзакта в модели есть время реакции моделируемой системы. Блок сбора статистики позволяет получить распределение времени реакции системы и определить его мат. ожидание и СКО.

В качестве единицы модельного времени выбрана 1 мс.

**Выводы.** Таким образом, разработаны аналитическая и имитационная модели расчета времени реакции распределенной вычислительной системы с резервированием коммуникационной среды. Аналитическая модель позволяет быстро получить грубую оценку времен реакции, для точных расчетов необходимо использование имитационной модели.

## **ИСПОЛЬЗОВАНИЕ МЕХАНИЗМА БИЗНЕС-ПРОЦЕССОВ ДЛЯ АВТОМАТИЗАЦИИ УЧЕТНЫХ ЗАДАЧ В ПРОГРАММЕ «1С: ПРЕДПРИЯТИЕ 8.0»**

Как известно для корректной с точки зрения бухгалтерского учета, регистрации хозяйственной операции в среде «1С: Бухгалтерия 7.7», практически всегда требуется ввести не один, а несколько документов, с помощью которых она регистрируется, формируются бухгалтерские проводки и т.д. Практическая работа бухгалтера, ведущего учет с помощью «1С: Бухгалтерия 7.7», сводится к вводу конкретной последовательности документов для регистрации той или иной хозяйственной операции. Такие последовательности документов часто называют «цепочками документов». По истечению достаточного времени работы с программой упомянутые «цепочки документов» в отношении постоянно регистрируемых этим бухгалтером хозяйственных операций становятся для него привычными, составляют его «трудовые будни». Тем не менее, можно отметить следующие недостатки такой технологии ведения учета в программе:

- пользователь должен самостоятельно «дойти до таких кондиций» работы с программой;
- контроль за выполнением отдельных звеньев цепочки целиком возлагается на пользователя программы;
- цепочки не включают в свой состав обязательные этапы выполнения операции, но не вызывающие изменения бухгалтерских итогов (например, получение разрешения от руководителя на «эксклюзивную» скидку при продаже партии товара, и т.п.).

Разработчики новой платформы программы, а именно, «1С: Предприятие 8.0» пошли по пути её совершенствования по целому ряду направлений (см., например [1]), в том числе и в плане повышения эффективности работы с программой в части автоматизации ввода и регистрации хозяйственных операций предприятия. Для этой цели в программу был добавлен так называемый механизм бизнес-процессов ([2]). Этот механизм включает средства для описания схем бизнес-процессов, их маршрутизации для

формирования заданий, выполняющихся в каждой точке маршрута, для управления бизнес-процессом и организации его связи с другими функциями прикладного решения. Рассмотрим этот механизм подробнее.

Механизм бизнес-процессов – «work-flow!» в англоязычной литературе – позволяет объединять отдельные операции в цепочки взаимосвязанных действий, приводящих к достижению требуемого результата. Он позволяет формализовать процедуры обработки тех или иных событий возникающих в деятельности организации, с возможностью привлечения для этого одного или нескольких сотрудников предприятия. Во многих существующих информационных системах для решения задач work-flow используются специализированные продукты, которые приходится интегрировать с приложениями, решающими экономические задачи. В платформе «1С: Предприятие 8.0» механизм бизнес-процессов является частью платформы, его использование совершенствует «традиционную» технологию ведения учета с помощью отдельных документов, объединяя их в конкретные бизнес-процессы.



**Рис. 1.** Схема взаимодействия объектов механизма бизнес-процессов

Механизм бизнес-процессов (МБП) обеспечивается сразу несколькими объектами конфигурирования: бизнес-процессы, задачи, параметр сеанса, регистры сведений и справочники. При этом основные два первых объ-

екта конфигурации для МБП используют друг друга, остальные три объекта являются вспомогательными (см. рис. 1).

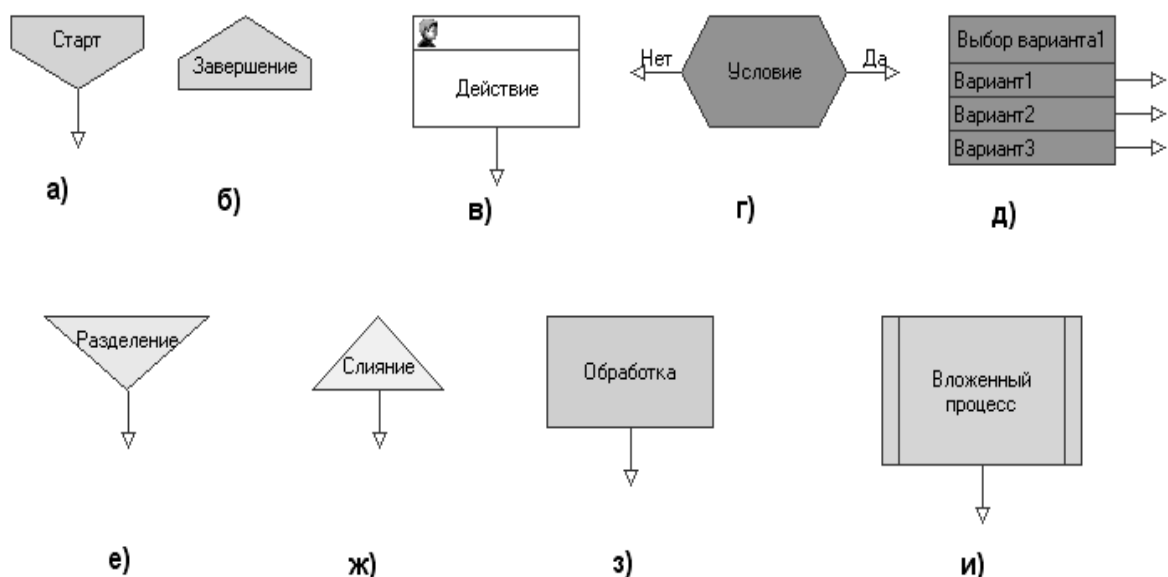
Задачи сообщают бизнес-процессам о своем выполнении, чем вызывают их продвижение дальше по маршруту. Регистр сведений используется ими для отбора задач для текущего исполнителя в соответствии с установленным параметром сеанса. Справочники используются при формировании задач вне бизнес-процессов (например, вручную) или, например, отборе задач.

Объект «Задача» предназначена для учета заданий и описывает способ их распределения по исполнителям, с учетом организационной структуры предприятия. Исполнителями заданий (адресация заданий) могут быть как конкретные сотрудники предприятия, так и обезличенные сотрудники (ролевая адресация). Так же задание может быть адресовано целым рабочим группам, подразделениям, помещениям, филиалам и т.д. При этом задачи могут создаваться не только бизнес процессами, но и другими объектами информационной базы и непосредственно пользователями. Более того, в общем случае исполнителем задания может быть не только сотрудник, но и любая внешняя система, например, другая учетная система.

Понятие задачи фактически определяет лишь интерфейс взаимодействия бизнес-процесса с заданием, выполнение которого может быть, в общем, случае не связано с выполнением операций в самой системе. Например, бизнес-процесс по ходу своего выполнения может потребовать согласования какого-то вопроса с руководителем фирмы. Сформулированная таким образом задача будет, к примеру, адресована секретарю, который будет решать ее любыми доступными ему способами: по электронной почте, по телефону и т.д. Задача будет считаться выполненной, когда в систему поступят сведения о получении нужного согласования. Каждая выполненная задача «продвигает» бизнес-процесс к выполнению следующего шага.

Объект «Бизнес-процесс» описывает логику выполнения операции для достижения той или иной цели и управляет жизненным циклом созданных бизнес-процессов (экземпляров) от момента старта до момента завершения. По аналогии с другими объектами метаданных (например, справочниками или документами) бизнес-процессы могут содержать различные реквизиты, табличные части и формы для взаимодействия с пользова-

телем. Отличительная особенность и в то же время ключевая характеристика бизнес-процессов — карта маршрута, которая содержит схематическое описание бизнес – логики, представляющий собой алгоритм выполнения отдельной хозяйственной операции средствами конфигурации «1С: Предприятие 8.0». Карты маршрута позволяет визуально описывать маршрут бизнес-процесса в виде связанного графа и позволяет легко описывать алгоритмы условных переходов, и реакцию бизнес-процесса на различные события. При работе пользователя с прикладным решением предусмотрена возможность отображения актуальной карты маршрута для конкретных экземпляров бизнес-процессов с учетом пройденных и активных точек маршрута. Для графического представления карты маршрутов используются интуитивно понятные графические фигуры и обозначения, учитывающие требования ГОСТ 19.003-80 «Схемы алгоритмов и программ. Обозначения условные графические» применяемые для представления алгоритмов решения различных задач в виде блок – схемы. Процесс составления карты маршрута состоит в «нанесении» на карту точек маршрута и определения связей между ними. Название точек маршрута, их графические изображения, используемые при создании карт маршрутов различных бизнес-процессов представлены на рис. 2.



**Рис. 2.** Виды точек маршрута

( а) – старт; б) – завершение; в) – действие; г) – условие; д) – выбор варианта;  
 е) – разделение; ж) – слияние; з) – обработка; и) – вложенный процесс)



Следует отметить, что по умолчанию создание упомянутого выше объекта «Задача» происходит только при переходе на такие точки маршрута как «Действие» и «Вложенный процесс». В остальных точках может производиться какая-либо обработка данных: проверка условия, выбор варианта движения по карте маршрута и т.п.

Бизнес-процессы в «1С: Предприятие 8.0» допускают несколько видов маршрутизации:

- **жесткая.** Бизнес-процесс имеет карту с жестко определенными адресатами для каждой точки маршрута, все действия бизнес-процесса выполняются последовательно друг за другом;

- **свободная.** Адресаты точки карты маршрута бизнес-процесса не установлены и определяются программно или интерактивно в течение жизненного цикла бизнес-процесса;

- **условная.** Выбор следующего действия в карте маршрута (ветви маршрута) выполняется в зависимости от выполнения, или невыполнения некоторого логического условия (бинарный переход). Выбор варианта развития бизнес-процесса может быть также множественным, т. е. не их двух, а нескольких ветвей маршрута;

- **параллельная.** Карта маршрута предусматривает разделение бизнес-процесса на параллельные ветви с возможностью последующего слияния (ожидания). Продвижение бизнес-процесса по каждой из параллельных ветвей происходит независимо по мере выполнения соответствующих задач.

Из сказанного можно сделать вывод о том, что виды карт маршрутов соответствуют базовым алгоритмическим структурам, в частности, можно отметить линейные и разветвляющиеся алгоритмы.

На рис. 3. представлена карта маршрута бизнес-процесса «Продажа товара».

Операции, выполняемые в ходе бизнес-процесса, представлены на карте маршрута точками действий (рис. 3), которые содержат информацию о том, кто и что должен сделать на данном этапе. Например:

- менеджер – выписать счет поставщику, оформить накладную;
- кассир — принять оплату наличными;
- кладовщик — произвести отгрузку товара со склада по накладной.

Исполнитель может определяться персонально (Иванов), или с учетом ролевой маршрутизации («Кладовщик», «Руководитель отдела продаж»). При переходе бизнес-процесса на точку действия он автоматически формирует задачу, устанавливая в них предусмотренные реквизиты адресации. После того, как исполнитель отметит задачу как выполненную, бизнес-процесс автоматически переходит к следующей точке маршрута в соответствии с картой.

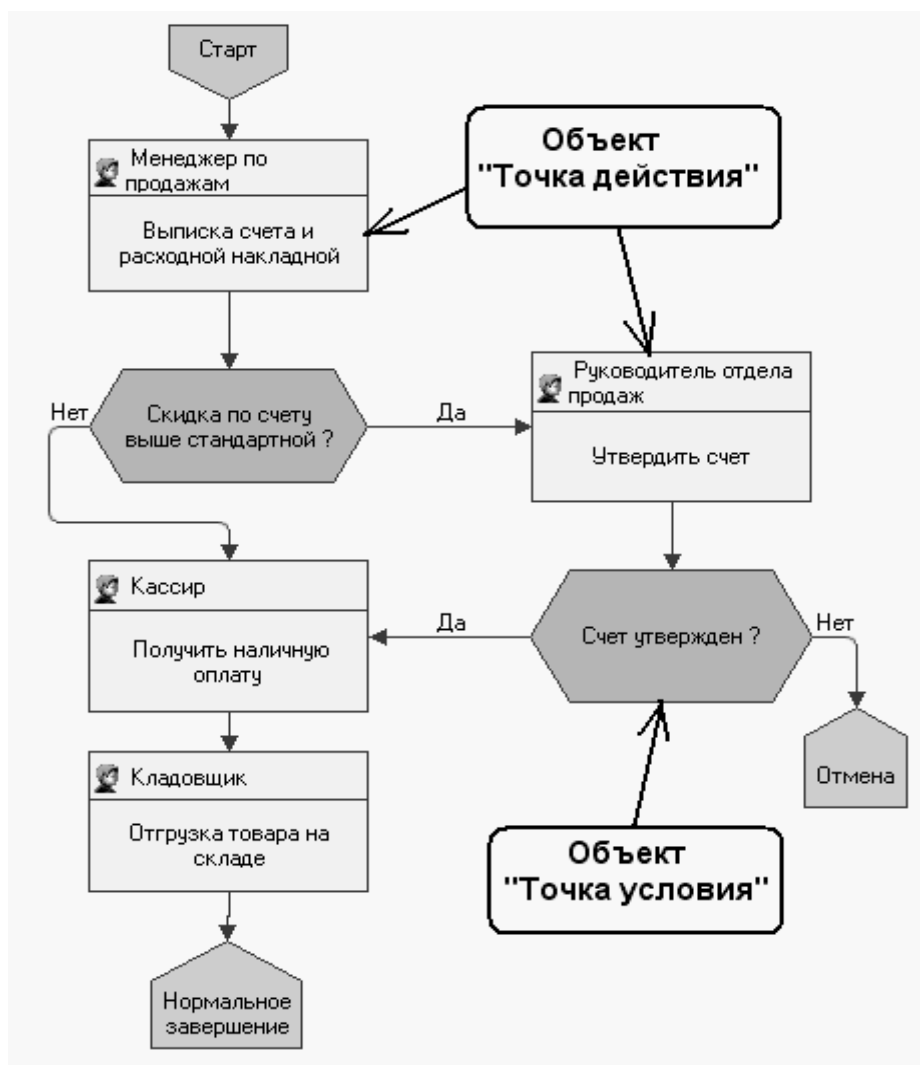


Рис. 3. Карта маршрута бизнес-процесса

Проверка условий в соответствующих точках карты маршрута (см. рис. 3) требует написания программного кода. Для программирования бизнес-процессов на платформе «1С: Предприятия 8.0», используется механизм, автоматизирующий эту процедуру за счет визуальных средств про-

ектирования, сводя к минимуму (или вовсе исключая) объем кодирования. Все это реализовано на уровне платформы, которая содержит объекты метаданных и механизмы, делающие возможным единообразную реализацию бизнес-процессов в прикладных решениях.

Самая очевидная польза от автоматизации бизнес-процессов заключается в том, что система сама начинает следить за регламентом выполнения работ, снимая это бремя с сотрудников и тем самым, высвобождая ресурсы для основной работы. Как следствие, сокращается число ошибок и повышается эффективность труда. В рамках разработанного бизнес-процесса может безошибочно работать даже неопытный пользователь, так как бизнес-процесс сам «ведет» пользователя, контролируя прохождение его этапов. Необходимо также отметить, что эффективность взаимодействия повышается за счет автоматической передачи информации сотрудникам по мере продвижения бизнес-процесса по маршруту: по оценкам специалистов, на взаимодействие сотрудников (передача и поиск информации) тратится от 20 до 40% рабочего времени [3].

Механизм бизнес-процессов предлагает готовую стратегию автоматизации совместной деятельности работников предприятия. Положительный эффект от применения механизма бизнес-процессов виден не только при эксплуатации системы, но и на этапе выработки проектных решений. Так, появляется возможность еще при проектировании сделать акцент на управленческий учет вообще и процессное управление в частности. В результате готовый продукт можно демонстрировать заказчику в виде простых и понятных схем бизнес-процессов, придающих жесткий каркас универсальному прикладному решению.

## ЭЛЕКТРОННЫЕ УЧЕБНЫЕ ПОСОБИЯ

В последние годы ведущие вузы страны уделяют все большее внимание созданию эффективных электронных учебных пособий (ЭУП) (иногда называемых электронными образовательными ресурсами, автоматизированными учебными курсами или просто электронными учебниками (ЭУ)).

В соответствии с мировым опытом на смену текстографическим электронным продуктам приходят высоко интерактивные, мультимедийно насыщенные ЭУП. При этом необходимо обеспечить возможности сетевого распространения, поскольку в географических условиях нашей страны телекоммуникационный доступ к образовательным ресурсам трудно переоценить.

Особенно важны требования к интерактивности и мультимедийной насыщенности для учебных продуктов, используемых в системе высшего заочного, дополнительного и дистанционного образования.

Использование ЭУП в образовательном процессе позволяет более углубленно изучить материал, ознакомиться более подробно с интересующими или трудными темами. Богатый и красочный иллюстративный материал позволяет наглядно продемонстрировать теоретическую информацию. При использовании электронных учебных пособий происходит не только репродуктивная деятельность, но и абстрактно-логическая, что способствует лучшему усвоению материала. Очень важен тот факт, что студент имеет возможность и на лекции, и на практических занятиях, и на самостоятельной работе пользоваться одним и тем же курсом, разработанным одним или несколькими авторами, при этом формируется целостный образ изучаемого предмета с его особенностями и нововведениями.

Таким образом, студент не только может самостоятельно изучить теоретический материал, подробнее остановившись на наиболее трудном или интересном непосредственно для него, но и может проработать практические задания по необходимой теме.

ЭУП – в большей степени инструмент обучения и познания, и его структура и содержание зависят от целей его использования. Он и репети-

тор, и тренажер, и самоучитель. Компьютерный учебник представляет собой замкнутый курс, не требующий, в принципе, обращения ни к какой книге.

Центральной идеей было соединение преимуществ книги и лекции: студент может выбирать собственную скорость изучения материала, в любой момент вернуться назад или найти ссылку на любой, интересующий его вопрос темы. В то же время последовательное появление на экране текста, графических образов и «фильмов» поможет задать должный ритм изучения.

Материал в ЭУП должен излагаться «прерывисто». Каждый раздел должен завершаться вопросами для самопроверки или тестами, с помощью которых реализуется обратная связь в процессе обучения. Электронный учебник эффективен только при наличии обратной связи; возможности быстрого поиска необходимой справочной информации; демонстрационных примеров и моделей (учебник рассказывает, показывает, объясняет, демонстрирует); когда организован контроль знаний (тренажер, самоконтроль, тестирование). Сочетание этих элементов обеспечивает более быстрое и долговременное усвоение материала электронного учебника.

Всё ЭУП состоит из разделов, имеющих соответствующие подразделы. Каждый раздел предваряет структурно-логическая схема, содержащая информацию об основных блоках материала, предлагаемого для изучения. На схеме стрелками указан желательный порядок изучения тем, но студент имеет возможность самостоятельно выбрать тему для изучения.

Минимальное, с технологической точки зрения, требование к созданию современного бумажного учебника – это последовательное, линейное изложение материала. В отличие от классического варианта учебника, электронный учебник предназначен для иного стиля обучения, в котором материал представлен в интерактивном изложении учебной информации, в виде так называемого *гипертекста*, снабженного взаимными ссылками на различные части материала учебника.

Учебно-информационный текст электронного учебника должен быть четко иерархически структурирован по содержанию. Базовый уровень иерархии отражает основные понятия и концепции предметной области. Более высокие уровни должны последовательно детализировать и конкретизировать эти понятия.

Важнейшей частью учебника должен являться глоссарий или структурированный каталог понятий и модулей. С помощью этого обеспечивается навигация по электронному учебнику.

ЭУП обычно имеет графический интерфейс пользователя и общепринятые в такого рода системах средства навигации и подсказки. Такое пособие, безусловно, может быть использовано на аудиторных занятиях (при наличии соответствующего оборудования для проецирования изображения на экран), но оно обладает и достаточными сервисными функциями для самостоятельного изучения.

Самостоятельное обучение должно подкрепляться возможностями самодиагностики знаний и контроля (измерения) обучения. Поэтому необходимо, чтобы в учебнике были контролирующие и тестирующие элементы.

Различают несколько составляющих в ЭУП. Основная – это **информационный блок**, который должен содержать изложенный в сжатой форме учебный материал. Каждый раздел информационного блока должен заканчиваться контрольными вопросами, которые позволяют обучаемому выяснить, насколько глубоко он усвоил учебный материал. В результате в электронном учебнике функционирует постоянная обратная связь обучаемого с «электронным преподавателем», позволяющая повысить эффективность процесса усвоения знаний.

После выбора ответа студентом, независимо от того, правильный был выбран ответ или нет, есть возможность познакомиться с решением этой задачи, что помогает глубже осмыслить изучаемый материал.

Блок **практические задания** должен функционировать в режиме диалога обучаемого с «электронным преподавателем». Работа в этом режиме дает возможность обучаемому закрепить знания, полученные при работе с информационным блоком. Система подсказки, при этом, позволяет при необходимости обратиться к любому разделу учебного материала. При составлении заданий этого блока должны использоваться различные системы конструирования ответа.

При этом необходимо особое внимание уделить демонстрации такого учебного материала, который является труднодоступным для обучаемого. Речь идет, в первую очередь, о демонстрации процессов с использованием средств мультимедиа и видеотехники в динамическом режиме на кон-

кретных примерах. В курсе информационных технологий, например, можно организовать просмотр технологии выполнения определенных операций в различных приложениях, демонстрацию выполнения процедур и получение результатов и т.д.

Контроль знаний обучаемых должен осуществляться в блоке **тест**. В этом блоке должна быть предусмотрена оценка правильности ответов обучаемого на поставленные вопросы. В конце теста обучаемому должна быть представлена информация о его результатах. Для устранения возможности запоминания обучаемым правильных ответов необходимо использовать значительный по величине банк компьютерных тестовых заданий, в котором варианты заданий перемешиваются и берутся в виде произвольной выборки.

Существуют два вида электронных учебников: закрытый электронный учебник и Internet-учебник. Первый представляет уже ставший традиционным учебник по предметной области, которой является независимым и неизменяемым. Он используется на персональных компьютерах или в локальных компьютерных сетях и распространяется на дискетах или компакт-дисках. Под Интернет-учебником понимается открытый и имеющий ссылки на внешние источники информации, базы данных и знаний электронный учебник, размещаемый на одном из серверов глобальной компьютерной сети.

Чтобы стать конкурентоспособным специалистом на рынке труда студент должен получить полное представление о современных достижениях в данной предметной области. Для решения этой задачи должна быть разработана такая структура и техническая реализация электронного учебника, которая обеспечила бы оперативную модификацию его содержания и дополнение новыми данными, включение новых гиперссылок на справочные библиографические данные, причем эта работа должна выполняться непосредственно составителями учебников без помощи программистов.

В связи с переходом на новые принципы организации обучения, когда на самостоятельное изучение материала студентом отводится 50 процентов времени, отводимого на данный курс, развитие системы дистанционного и открытого обучения представляется единственно возможным и весьма перспективным. Разработка и поддержание в актуальном состоянии мето-

дически выверенных, разнообразно оформленных и дружественных обучающих сред, называемых интерактивными компьютерными или электронными учебниками позволит студентам самостоятельно изучать курс.

На сегодняшний день имеется большое число ЭУ. Но в основном они представляют собой простое отображение текста учебника в электронном виде, с навигацией по тексту и имеют ограниченные возможности для работы с учебником. Есть учебники, которые разработаны более тщательно, имеющие различные мультимедийные средства: видео, аудио, анимацию. Также в них присутствуют различные тренажеры и тесты для самоконтроля студента. Но в них не предусмотрены возможности добавления нового материала.

Существуют различные методики создания электронных учебников. В них описываются поэтапное создание основных модулей: оглавление, предметный указатель, тесты, а также разбиение текста учебника на главы, разделы и т.д. для удобной работы с материалом. Некоторые методики предлагают использовать различные программы для разработки электронных учебников.

Но эти методики имеют два недостатка: не предусматривают создание полного перечня возможностей электронного учебника; ссылаются на программы, для создания ЭУ, которые требуют много ручной работы.

Специалисты ведущих российских университетов, много и плодотворно занимающихся разработкой электронных учебников (Московский государственный институт электроники и математики (технический университет), Томский Политехнический Университет, Кемеровский государственный университет, РЭА им. Г.В. Плеханова и др.) проведя анализ существующих ЭУ сформулировали основные модули, из которых должен состоять ЭУ:

**Оглавление.** Один из важнейших модулей. В оглавлении должны быть ссылки на все ресурсы (текст, тесты, тренажеры и т.д.) ЭУ. Желательно, чтобы оглавление имело древовидную структуру, т.е. оно должно содержать папки, подпапки и т.д.

**Предметный указатель.** В этом разделе должны содержаться основные термины со ссылками на определение и местами упоминания их в тексте. Для удобного поиска нужного термина они должны быть отсортированы по алфавиту и объединены в папки по первой букве термина.



**Тесты.** Позволяют пользователю оценить уровень знаний. Тесты могут быть разных типов: выбрать правильный ответ из списка предоставленных, вписать нужное слово и т.д.

**Тренажер.** Это программа, которая позволяет студенту виртуально применить полученные знания, например при решении задач. Перед ним ставится задача, даются условия и предоставляются инструменты для её решения. Это эмитирует жизненную ситуацию и позволяет отработать навыки решения, что в дальнейшем поможет студенту справиться с ней в реальности.

**Ведение конспекта.** При работе с ЭУ студенту предоставляется возможность записывать важную информацию со своими комментариями в виде конспекта, добавлять ссылки на те места в учебнике, где это упоминается. В последствии, он сможет обратиться к своим записям для быстрого поиска нужной информации.

**Ведение статистики.** Система фиксирует время, потраченное студентом на изучение каждой главы учебника, чтобы определить, какому разделу он уделил мало времени или узнать, что ему ещё необходимо прочитать. Также преподаватель сможет определить тему, которую студент знает плохо, а какую вообще не читал.

**Контроль внимания студента.** Программа позволяет определить как внимательно пользователь работает с текстом. Например, искажается какое-нибудь слово. Если студент внимательно читает текст, то он заметит ошибку, нажмет на это слово, и оно изменится на правильное, а программа зафиксирует это событие. Или другой вариант: через определенный промежуток времени текст учебника будет исчезать, и если студент читает текст, то, переместив мышку, текст вернется, а если студента нет на месте, то время, когда текст был невидим, не будет учтено.

Вышеназванная методика подразумевает, что для массового перевода учебников в электронный вид, разработчикам необходим комплекс программ, по созданию оглавления, предметного указателя, тестов и формы для работы с электронным учебником, которые работают автоматически с минимальным вмешательством человека.

В настоящее время многие компьютерные фирмы и организации занимаются разработкой *электронных учебников*.

При этом некоторые фирмы-разработчики идут по пути *создания учебников с помощью распространенных WWW-технологий*. Их электронные учебники представляют из себя сложным образом организованный гипертекст с включенными в него иллюстрациями, аппаратом контроля знаний, справочными материалами и, при необходимости, компьютерными моделями и симуляциями, позволяющими, помимо теории, провести серию экспериментальных и практических работ.

Создание подобных электронных учебников сопряжено с большими затратами времени и средств. Не всегда удается организовать и проконтролировать работу с данным учебником большого количества обучаемых. Зато каждый учебник, созданный с помощью «ручного» труда, становится таким же уникальным, как и каждый хорошо изданный традиционный учебник «на бумаге».

Другие фирмы идут по пути создания мощных по своим возможностям *инструментальных оболочек*, с помощью которых можно в довольно сжатые сроки создать большое количество электронных учебников, наполняя «пустую» оболочку текстовыми и графическими материалами. Современные оболочки предусматривают возможность доступа к внешним ресурсам Интернет (выход в поисковые системы и базы данных, работу с тематическими сайтами), участия в телеконференциях и чатах с преподавателями. Значительное место в подобных учебниках занимает тестирование. Очень важным является то, что при использовании инструментальных оболочек удается организовать одновременную работу с ними большого количества обучаемых, что является затруднительным при использовании WWW-технологий. В настоящее время подобные инструментальные оболочки создаются *практически в каждом российском университете*, занимающемся развитием системы дистанционного обучения с использованием Интернет-технологий (например, в известном Московском государственном университете экономики, статистики и информатики, МЭСИ – <http://www.ido.ru>). Также созданием подобных инструментальных оболочек-программ занимаются уже несколько лет такие известные *фирмы-разработчики* программного обеспечения как Oracle, Lotus (<http://dlc.miem.edu.ru>), IBM, Maris Multimedia (<http://edugen.com>) и др.

При создании электронных учебников, размещаемых в Интернет, необходимо учитывать особенности, присущие всем электронным учебни-

кам плюс особенности телекоммуникационной среды. По мнению некоторых авторов, учебник для дистанционного обучения должен обладать следующими качествами:

- *развитой гипертекстовой структурой* в понятийной части курса (определения, теоремы), а также в логической структуре изложения (последовательность, взаимозависимость частей);

- *удобной для пользователя системой управления структурой* – преподаватель может задать любую форму представления и последовательность изложения материала, что позволяет один и тот же учебный материал использовать для аудитории разной степени подготовленности и для различных видов учебной деятельности;

- *использованием*, если это методически оправдано, *звука, анимации, графических вставок, слайд-шоу* и т.п. Учащийся также должен иметь возможность распечатать любую «страницу» подобного учебника;

- *доступностью* для ученика, причем по возможности, несколькими способами (например, и по Интернет, и на CD-ROM);

- наличием *подсистемы контроля знаний, интегрированной в учебник*;

- возможностью организовать *доступ непосредственно со страниц учебника к самым разным информационным ресурсам Интернет* и пр.

В Интернет-учебниках могут использоваться различные технологии представления информации. К настоящему моменту за основу представления информации во «Всемирной паутине» (WWW) взят *гипертекст*. В основе гипертекста лежит расширенная модель энциклопедии – веками отработанного средства информационной поддержки образования. Современная электронная энциклопедия, помимо фотографий, содержит звукозаписи, музыкальное сопровождение и видеофрагменты. *Модель энциклопедии* строится на следующих принципах:

- свобода перемещения по тексту;
- сжатое (реферативное) изложение информации;
- необязательность сплошного чтения текста;
- справочный характер информации;
- использование перекрестных ссылок.

Эта модель близка к учебным видеофильмам, однако, их применение в дистанционном обучении пока затруднено, так как для показа видеофильма нужно передавать по сети слишком большое количество информации.

Сочетанием динамики с разумными объемами передаваемой информации оказываются *компьютерные слайд-фильмы*. В отличие от обычных слайдов, они имеют средства квази-мультипликации, позволяющие существенно разнообразить динамику показа, слайды могут сопровождаться звуком, музыкой, содержать кинофрагменты. Модель компьютерных слайд-фильмов строится на принципах:

- динамика предъявления текста задается преподавателем (это происходит либо заранее при разработке фильма, либо в процессе демонстрации);
- допускаются перекрестные ссылки;
- логика предъявления текста задается преподавателем (этот процесс может иметь варианты-ветви);
- компьютерный слайд-фильм предназначен для сплошного просмотра;
- компьютерный слайд-фильм навязывает ученику свою логику изучения материала;
- компьютерный слайд-фильм задает ритм прохождения материала и имеет специальные аудиовизуальные средства управления восприятием материала;
- обучаемые могут разрабатывать фильмы сами, что создает хорошую техническую базу для применения проектного метода обучения.

Просмотрев большое количество существующих в настоящее время в Интернет электронных учебников, можно сделать вывод, что их структуры в основном базируются на привычной парадигме «книги» – некотором количестве иллюстрированной текстовой информации (в лучшем случае с небольшим количеством гиперссылок). Такая структура вполне подходит, если учебник служит вспомогательным материалом в процессе традиционного обучения, но явно недостаточна в случае обучения дистанционного, когда общение учителя и ученика сведено к минимуму.

К сожалению, многие курсы дистанционного обучения не предусматривают эффективной обратной связи с преподавателем (например, использование IRC-технологий), многооконную структуру курса, не уделяют достаточно внимания дизайну Web-страниц. Многие не предусматривают и общения с преподавателем (координатором) курса даже по электронной почте. Авторы дистанционных образовательных программ указывают на проблему отслеживания и контроля выполнения заданий учащимися, осо-

бенно в тренировочных упражнениях, где задержка информации о правильности/неправильности (при посылке выполненного упражнения координатору по электронной почте) или при выборе из нескольких вариантов ответов (без контроля со стороны куратора) неблагоприятно сказывается на качестве вырабатываемого навыка. Вместе с тем, на ряде серверов Интернет можно уже найти курсы, в которых интерактивность встроена на уровне технологического обеспечения, что, естественно, можно считать решением до определенной степени проблемы организации дистанционного обучения под руководством преподавателя. Будущее за такими курсами.

**ДЛЯ ЗАМЕТОК**

---

---

## Содержание

От редакции .....	1
<b>Заяц А. М.</b> Новый подход в практической подготовке студентов Лесотехнической академии.....	4
<b>Заяц А. М., Гоголевский А. С.</b> Автоматизация разработки лесохозяйственных регламентов и проектов освоения лесов на платформе виртуальных лесничеств.....	13
<b>Богатырев В. А., Богатырев С. В.</b> Векторная оптимизация структуры кластера.....	19
<b>Гуров С. В.</b> Эффективность функционирования динамической системы с дискретным множеством состояний .....	28
<b>Недорезов Л. В.</b> О некоторых ключевых проблемах экологического моделирования.....	37
<b>Панфилов И. В.</b> Проблемное обучение.....	49
<b>Терентьев И. В., Неверовский В. Ю., Шубина М. А.</b> Алгоритмы автоматического определения возраста растений и животных по фотографиям.....	52
<b>Терентьев И. В., Шубина М. А., Морус Г. А.</b> Усовершенствованный алгоритм классификации лесохозяйственных объектов на аэрокосмических изображениях земной поверхности.....	65
<b>Уткин Л. В., Нгуен Ван Х.</b> Пессимистический подход к многокритериальному принятию решений в рамках теории случайных множеств .....	75
<b>Васильев Н. П., Ковешников А. И., Пресняков В. А.</b> Вариант решения задачи экспорта данных .....	85
<b>Ганичев И. В., Троицкая Е. Г., Троицкая А. Г.</b> Системный анализ иерархической модели ЭВМ.....	94
<b>Горбачев В. А., Потокин А. Ф.</b> Создание базы данных электронного каталога коллекции «Гербария им. И. П. Бородина» СПбГЛТА. Акроним (индекс): kfta .....	104
<b>Кондрашкин В. Е.</b> Решения алгебраических и трансцендентных уравнений прикладными программами.....	110
<b>Коваленко Т. В.</b> Применение информационных систем для рациональной организации лесных грузопотоков.....	117
<b>Курилова М. Н.</b> Опыт проведения и участия в олимпиадах по информатике и программированию кафедры информатики и информационных систем .....	122
<b>Лебедев М. О.</b> Принципы создания динамически настраиваемых информационных систем .....	127
<b>Лушкин Н. В.</b> Управление процессом противоточного выщелачивания пульсациями растворителя.....	138
<b>Михайлов А. А.</b> Информационные технологии – организатор и катализатор образовательного процесса .....	143

<b>Панфилова Е. И.</b> Методика проектирования охранной системы защиты компьютерной системы предприятия .....	147
<b>Пресняков В. А.</b> Теория и практика БД в СПбГЛТА .....	157
<b>Симанова Н. В.</b> Групповое многокритериальное принятие решений на основе алгебры предпочтений .....	163
<b>Хабаров С. П., Колмогорцев Е. Л.</b> К вопросу об оценке времени реакции распределенной системы управления с резервированием сетевой коммуникационной подсистемы .....	173
<b>Халиков М. И.</b> Использование механизма бизнес-процессов для автоматизации учетных задач в программе «1С: Предприятие 8.0» .....	181
<b>Ярин Б. Д.</b> Электронные учебные пособия .....	188



*40 лет кафедре информатики и информационных систем*

**ИНФОРМАЦИОННЫЕ СИСТЕМЫ  
И ТЕХНОЛОГИИ:  
ТЕОРИЯ И ПРАКТИКА**

**Сборник научных трудов**

*Отпечатано в авторской редакции  
Компьютерная верстка Т. И. Балацкой*

---

Подписано в печать с оригинал-макета 22.09.08.

Формат 60×84/16. Бумага офсетная. Печать трафаретная.

Уч.-изд. л. 12,5. Печ. л. 12,5. Тираж 150 экз. Заказ № 217. С 248.

---

Санкт-Петербургская государственная лесотехническая академия

Издательско-полиграфический отдел СПбГЛТА

194021, Санкт-Петербург, Институтский пер., 5