

Министерство образования и науки РФ

Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ ЛЕСОТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ имени С. М. Кирова»

Кафедра информационных систем и технологий

ИНФОРМАЦИОННЫЕ
СИСТЕМЫ И ТЕХНОЛОГИИ:
ТЕОРИЯ И ПРАКТИКА

Сборник научных трудов

Выпуск 8

Санкт-Петербург
2016

Рассмотрен и рекомендован к изданию
институтом леса и природоведения
Санкт-Петербургского государственного лесотехнического университета
24 марта 2016 г.

Редакционная коллегия:

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

Составитель

М. А. Шубина, кандидат технических наук, доцент (отв. секретарь)

Рецензент

Доктор технических наук, профессор ВТ НИУИТМО **А. А. Ожиганов**

Информационные системы и технологии: теория и практика: сб. научн. тр. Вып. 8 / отв.ред. А. М.Заяц. – СПб.: СПбГЛТУ, 2016. – 128 с.

ISBN 978-5-9239-0839-8

Сборник подготовлен по материалам кафедры вуза, представленным на научно-технической конференции института леса и природопользования СПбГЛТУ в январе 2016г., и практических работ, выполненных ее сотрудниками.

*Гурову Сергею Владимировичу и
Преснякову Владимиру Александровичу*

ПОСВЯЩАЕТСЯ



Гуров Сергей Владимирович (13.08.1945 – 08.08.2015) – доктор технических наук, профессор, заслуженный работник высшей школы закончил математико-механический факультет Ленинградского государственного университета им. А.А.Жданова по специальности «математик». На кафедре Информационных систем и технологий С.В. Гуров начал работать с 1976г., а в 2001 году был приглашен на возрождающуюся кафедру Математических методов и компьютерных технологий (в настоящее время кафедра управления, автоматизации и системного анализа), но продолжал чтение лекций на нашей кафедре по дисциплинам "Обработка экспериментальных данных на ЭВМ", "Надежность технических систем и техногенный риск", "Моделирование информационных систем".

Из более чем тридцати учебников и учебных пособий С.В. Гурова важным, по его мнению, является учебник для вузов, который был написан им совместно со своим учителем доктором технических наук, профессором Половко Анатолий Михайловичем – «Основы теории надежности» и «Основы теории надежности. Практикум», изданные в 2006 году в издательстве ПХВ-Петербург.

Своими делами Сергей Владимирович Гуров внес неоценимый вклад в почти пятидесятилетнюю историю кафедры Информационных систем и технологий.



Пресняков Владимир Александрович

(07.03. 1952 – 01.01.2016) – кандидат физико – математических наук, доцент, закончил Ленинградский государственный университет по специальности «Прикладная математика».

Работал инженером в НПО «Ленинец» (1974 - 1977), обучался в аспирантуре ЛГУ и защитил диссертацию (1977 - 1979), исследователь в НИИ Вычислительной математики и процессов управления (1979 - 1985). С 1985 года начал свою педагогическую деятельность на кафедре Вычислительной техники и информатики ЛТА им. С.М. Кирова. Руководитель ИТ - фирмы (1991-2010). С 2010 доцент кафедры Информационных систем и технологий государственного лесотехнического университета.

Область деятельности Преснякова В.А. в должности доцента кафедры не ограничивалась только педагогической работой. Он активно занимался разработкой систем и информационных технологий, обеспечивающих автоматизацию учебного процесса в университете. Последний проект Владимира Александровича, по разработке информационно-образовательной среды на платформе виртуализации и облачных технологий с переводом всех информационных ресурсов на кластер серверов, в рамках модернизации информационной системы кафедры и сегодня продолжается его учениками и преподавателями - единомышленниками.

А. М. Заяц, кандидат технических наук, профессор
**ИТОГИ НАУЧНО – ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЫ И НИРС
2015 ГОД**

Коллектив кафедры в 2015 году продолжил проведение исследований в рамках направлений представленных на рис. 1, со следующими результатами (см. рис. 2 - 5).

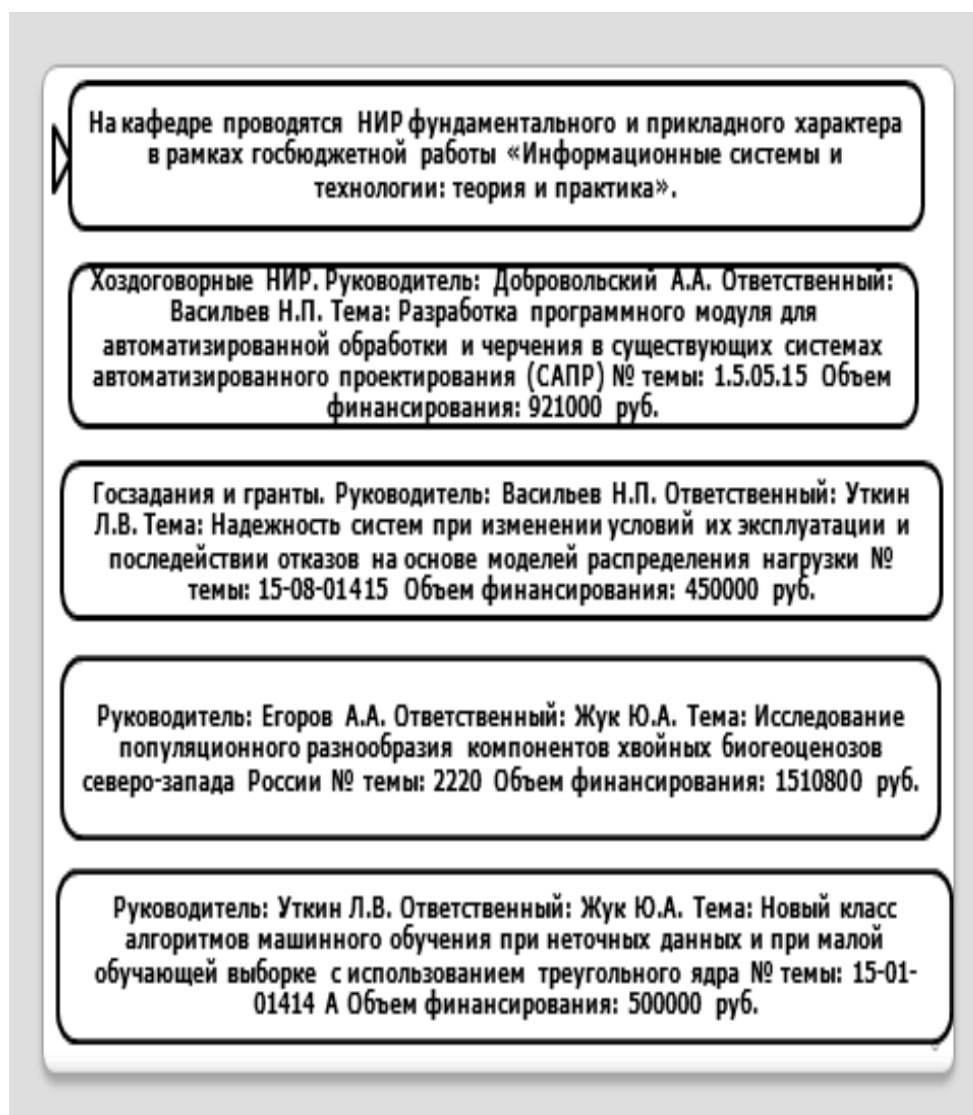


Рис. 1



Рис. 2

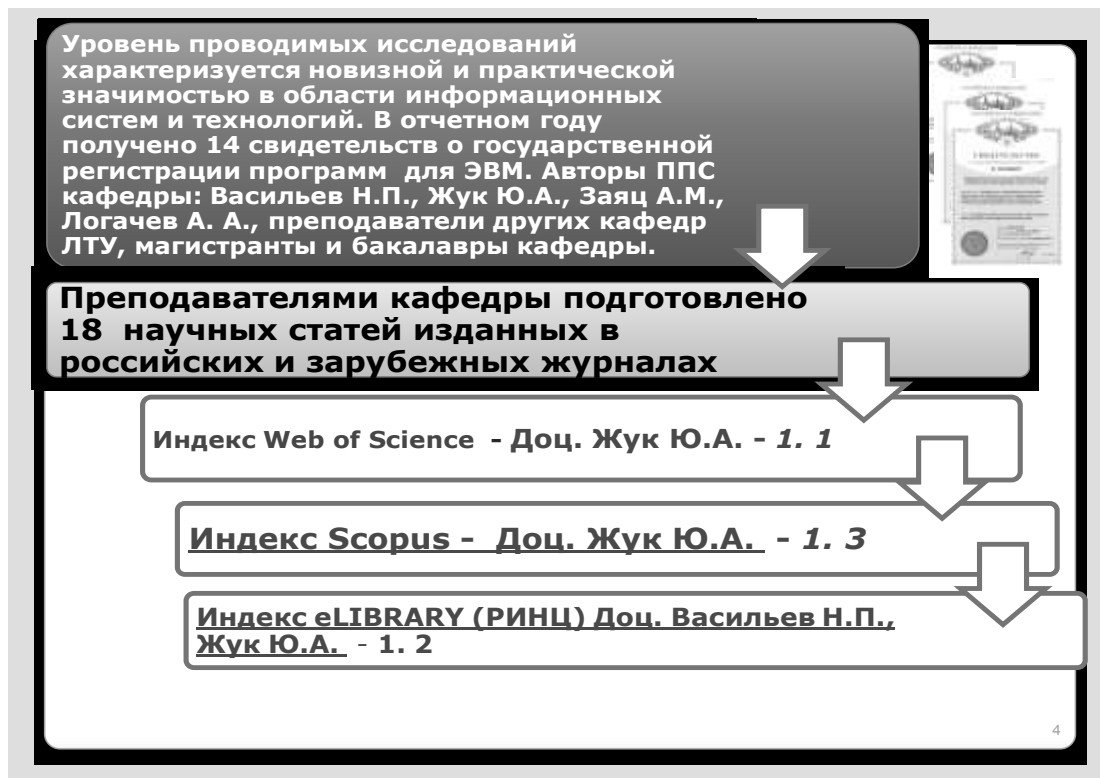


Рис.3

В отчетном году преподаватели кафедры заняли высокие места в различных рейтингах (см. рис. 4).

Доктор технических наук, профессор

Богатырев Владимир Анатольевич

Информатика: Топ-100 самых продуктивных российских учёных по данным РИНЦ – 31 место

Кибернетика: Топ-100 самых продуктивных российских учёных по данным РИНЦ – 41 место

Автоматика. Вычислительная техника: Топ-100 самых продуктивных российских учёных по данным РИНЦ – 71 место

5

Рис.4

За отчетный период на кафедре изданы следующие учебно-методические материалы.

Учебно-методические материалы, изданные в 2014/15уч. году

№ п/п	Название учебных, учебно-методических материалов	Изд-во, год	Авторы
1	Моделирование химико-технологических процессов. Учебное пособие	СПбГЛТУ 2015	Доц. Васильев Н.П.
2	Операционные системы. Учебное пособие	СПбГЛТУ 2015	Доц. Шубина М.А.
3	Интеллектуальные информационные системы Электр. Учебное пособие http://www.habarov.spb.ru/new_es/index.htm	2015	Доц. Хабаров С.П
4	Полезные утилиты и команды, используемые при работе в компьютерных сетях. Электр. Учебное пособие http://www.habarov.spb.ru/net_utility/index.htm	2015	Доц. Хабаров С.П

7

Рис.5

На кафедре организована научная работа студентов в рамках постоянно действующего кафедрального научного общества студентов (Руководитель ст. преподаватель Смелова Н.Б.). Итоги НИРС представлены на рис. 6.

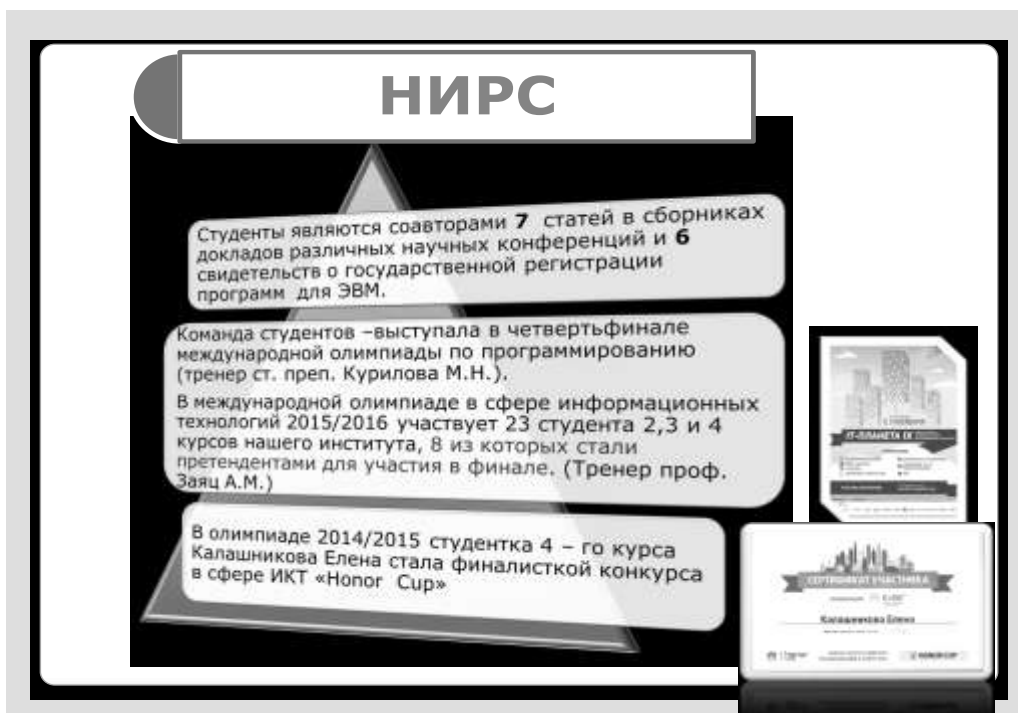


Рис. 6

Основные задачи кафедры на 2016 год представлены на рис. 7.

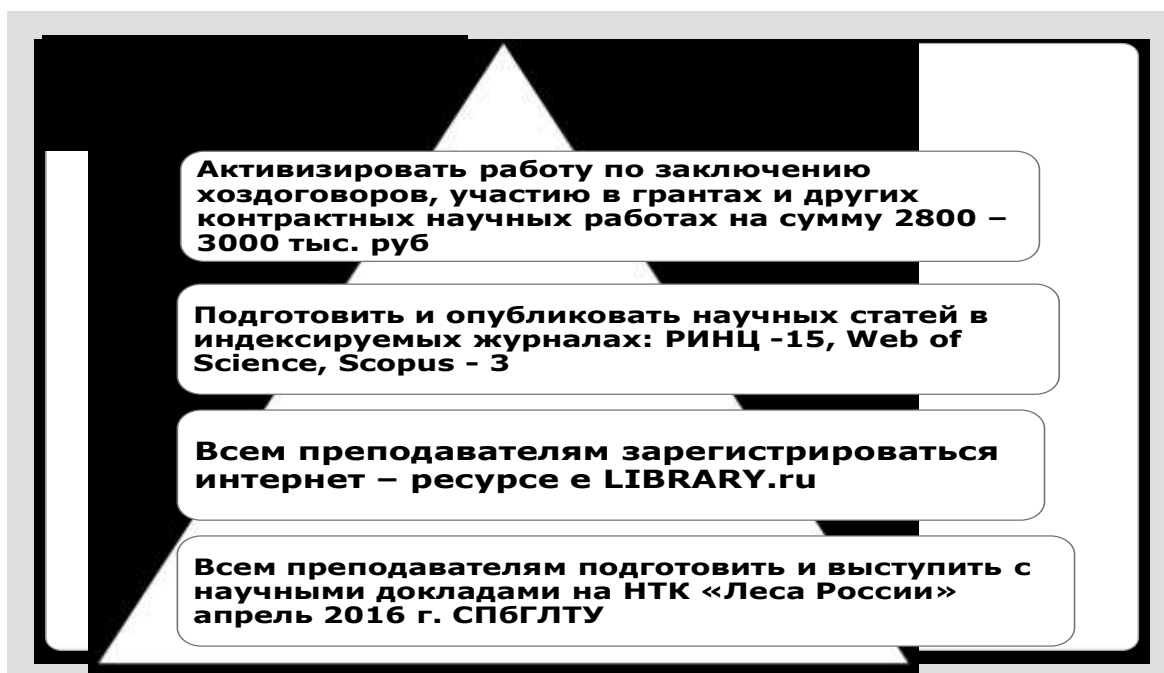


Рис.7

По итогам НИР за 2015 год в системе рейтинговой оценки коллектив кафедры занял 6 место из 49 кафедр СПбГЛТУ.

В рамках плана мероприятий на кафедре была организована и проведена очередная научно – практическая конференция *«Информационные системы и технологии: теория и практика»*, материалы докладов которой представлены в данном сборнике.

А.М. Заяц кандидат технических наук, профессор
М.И. Думов, студент 3 курса

ОБЗОР БЕСПРОВОДНЫХ СЕНСОРНЫХ СЕТЕЙ И ТЕХНОЛОГИЙ ИНФОРМАЦИОННЫХ СИСТЕМ ОЦЕНКИ ЛЕСНОЙ ПОЖАРООПАСНОСТИ И МОНИТОРИНГА ЛЕСОВ

Введение

Анализ информации по лесным пожарам (ЛП) [1,2,3] показывает, что они являются значительной экологической проблемой для всей территории России, где сосредоточена большая часть бореального пояса нашей планеты. Помимо губительных угроз жизни человека, отрицательных экологических последствий, лесные пожары наносят существенный экономический ущерб, связанный как со стоимостью самой древесины и недревесных лесных ресурсов, так и с организацией их обнаружения и тушения.

Одной из основных проблем, которую необходимо решить для снижения рисков и экономического ущерба от лесных пожаров, является обеспечение максимально раннего обнаружения пожара, своевременного и достоверного вычисления параметров его распространения [4].

Существующие системы обнаружения лесных пожаров, недостаточно эффективны, так как могут обнаруживать пожары не ранее чем на этапе воспламенения. Это связано с тем, что они основаны обычно на принципах визуального обнаружения воспламенения. Если же подобная система будет содержать в себе еще и как минимум сенсорную сеть, включающую в себя различные датчики, то она сможет обнаруживать возникновение пожара на более ранних стадиях, когда еще не причинен ущерб.

Краткая характеристика беспроводных сенсорных сетей

Анализ показывает [4,5,15], что одной из перспективных технологий получения достоверной и актуальной информации по возникновению и в дальнейшем по развитию ЛП является применение беспроводных сенсорных сетей (БСС, англ. WSN — Wireless Sensor Network). Сенсорные датчики БСС могут контролировать температуру и влажность воздуха, направление и скорость ветра, обнаруживать дым, открытый огонь и другие параметры, характеризующие лесные пожары. Так для оценки пожароопасности сенсор может содержать термодатчик, датчик влажности почвы и осадкометр осуществляющие фиксацию необходимых параметров для вычисления индекса Нестерова оценивающего степень горимости леса.

Совместное использование беспроводных сенсорных сетей и GPS, позволяет не только оперативно узнать о начале пожара, но и с высокой точностью определить, где находится очаг возгорания.

Беспроводная сенсорная сеть – это распределенная информационно – коммуникационная система необслуживаемых электронных устройств (уз-

лов сети) обладающих возможностями самоорганизации и осуществляющих сбор данных о параметрах внешней среды и их передачу на базовую станцию посредством ретрансляции от узла к узлу с помощью беспроводной связи [6].

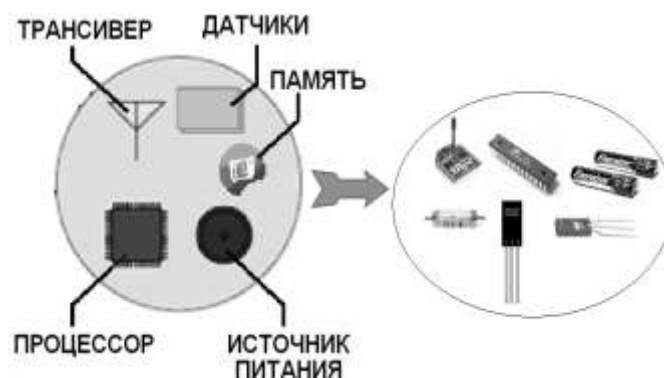


Рис.1. Состав узла сенсорной сети и его аппаратные компоненты

Узел сети (рис.1), называемый сенсором, обычно содержит датчик, воспринимающий данные от внешней среды (собственно сенсор), процессор, память, трансивер и автономный источник питания.

Такой набор позволяет устройству осуществлять измерения различных параметров внешней среды, самостоятельно проводить первичную обработку данных и поддерживать связь с другими информационными системами. Принципиальной отличительной особенностью беспроводных сенсорных сетей является принцип ретрансляции данных по цепи. Это позволяет собирать информацию со значительных площадей, превосходящих по своим размерам зоны связи одного устройства.

В настоящее время известны различные топологии БСС (см. рис.2).

Обычно беспроводные сенсорные сети строятся на основе протоколов 802.15.4, ZigBee, 6LoWPAN и DigiMesh [6,7,8].

Сети стандарта IEEE 802.15.4

В стандарте *IEEE* 802.15.4 выделяется два вида узлов БСС: устройства с ограниченными функциями RFD (reduced-function device), осуществляющие сбор информации в некоторой окрестности точки своего размещения и полнофункциональные устройства FFD (full-function device), выполняющие как ретрансляцию пакетов, так и сбор информации. Сенсорные сети в стандарте *IEEE* 802.15.4 могут включать 2^{64} сетевых устройств.

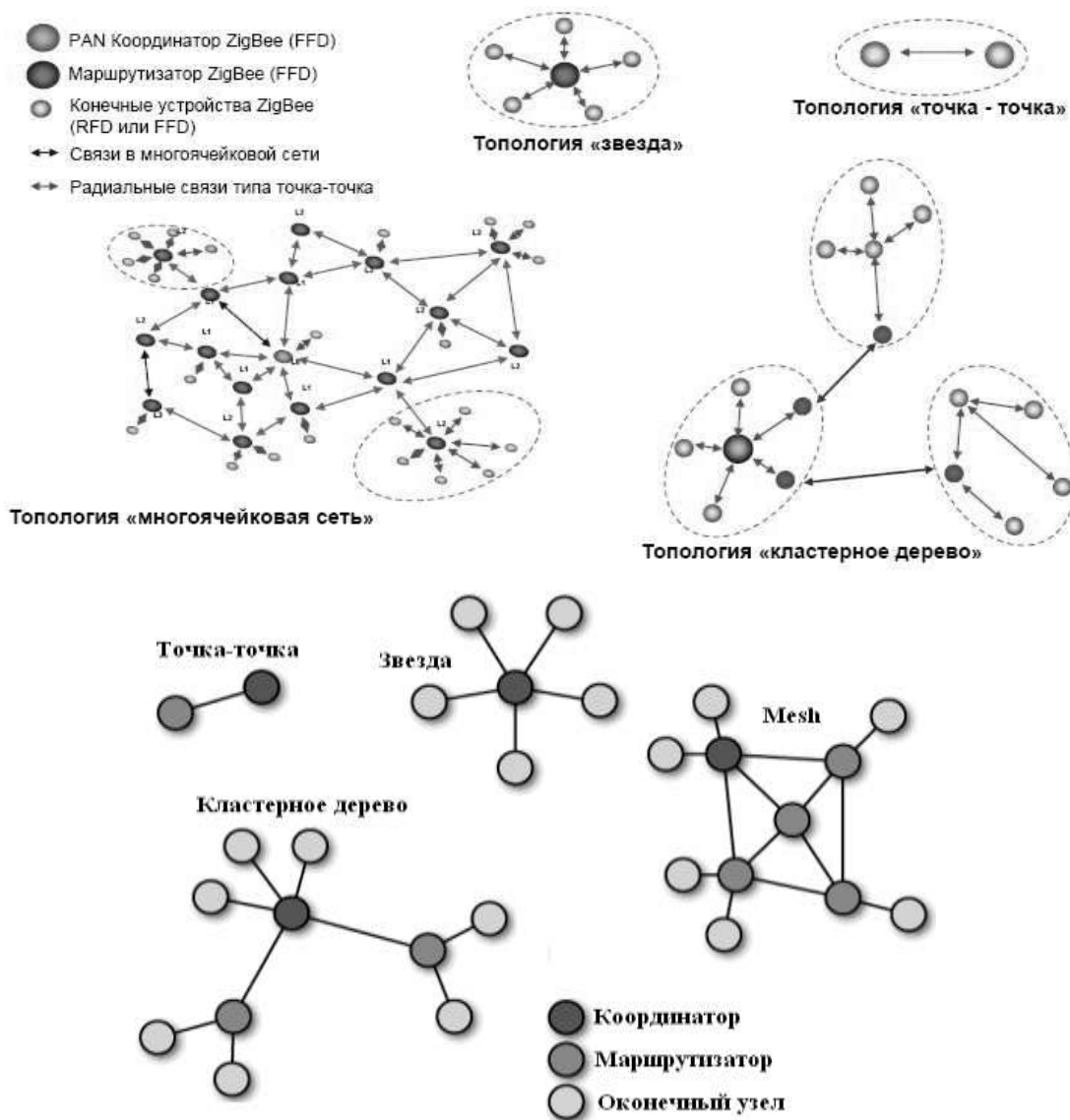


Рис.2. Поддерживаемые топологии беспроводных сенсорных сетей

В каждой сети стандарта *IEEE* 802.15.4 должно быть, по меньшей мере, одно полнофункциональное устройство для работы в качестве координатора.

Одноранговые сети могут формировать произвольные топологические структуры, ограниченные только дистанцией между каждой парой узлов. Примером одноранговой или пиринговой сети (peer-to-peer, *P2P* — равный к равному) является кластерное дерево. Сеть типа кластерное дерево является частным случаем сети *P2P*, в которой большинство устройств являются *FFD*. Устройства *RFD* подключаются к кластеру в качестве конечных узлов. Для присоединения к сети удалённых от координатора PAN (Personal Area Network) новых сетевых устройств могут использоваться уже присоединённые к сети *FFD* в режиме координатора. В этом режиме они, как и изначально координатор PAN, «зывают» маяками в сеть новые сетевые устройства. В результате формируется кластер из сете-

вых устройств, которые «слышат» своего координатора. Тем не менее, вся информация о кластере доступна координатору PAN. Подобным образом могут формироваться мультикластеры из сетевых устройств.

Стандарт *IEEE 802.15.4* является базовой основой для более высокоуровневых протоколов (*ZigBee*, *DigiMesh*, *6LoWPAN* и др.) позволяющий реализовывать с помощью программных надстроек на сетевом уровне и выше любую из топологий приведенных на рисунке 2.

Сети ZigBee

Спецификация *ZigBee* при малом энергопотреблении поддерживает не только простые топологии сети («точка-точка», «дерево» и «звезда»), но и самоорганизующуюся и самовосстанавливающуюся ячеистую (mesh) топологию с ретрансляцией и маршрутизацией сообщений. Здесь узел самостоятельно определяет свое местоположение, по крайней мере по отношению к узлу, которому он будет передавать данные. То есть сначала происходит идентификация всех узлов, а затем уже формируется схема маршрутизации.

Кроме того, ZigBee включает возможность выбора алгоритма маршрутизации, в зависимости от требований приложения и состояния сети, а также обеспечивает гибкий механизм безопасности, простоту развертывания, обслуживания и модернизации.

Стандарт ZigBee также поддерживает сеть с кластерной архитектурой. Кластер образуют маршрутизатор и конечные узлы, у которых он запрашивает сенсорные данные. Маршрутизаторы кластеров ретранслируют данные друг другу, и в затем данные передаются координатору. Координатор обычно имеет связь с IP-сетью, куда и направляются данные для окончательной обработки. В сетях ZigBee - дальность действия на разрешенной мощности в помещении – до 90 м, в прямой видимости – до 4000 м между соседними узлами сети.

Функционально сенсорная сеть ZigBee содержит узлы трех видов (см. рис.3):

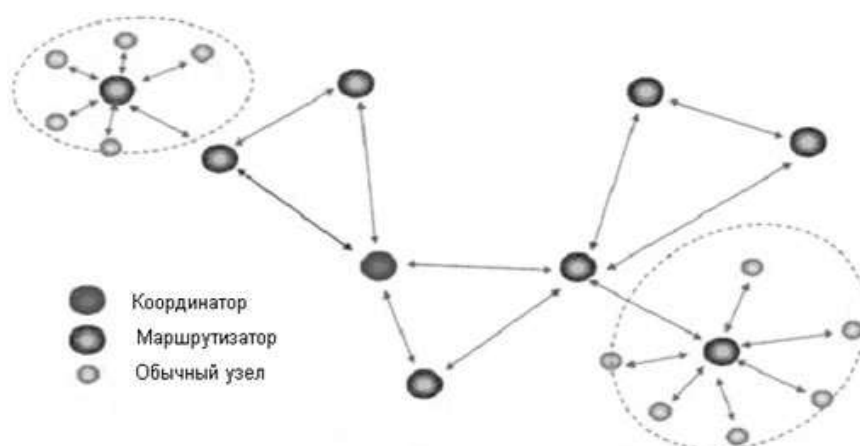


Рис.3. Произвольная структура беспроводной сенсорной сети ZigBee

1. ● Базовый узел или сетевой координатор (NCD — Network Coordination Device с дополнительными системными ресурсами в зависимости от сложности сети), осуществляет координацию, организацию и установку параметров сети. Координатор управляет работой сети, хранит данные о ее топологии и служит шлюзом для передачи данных, собираемых всей беспроводной сенсорной сетью, для дальнейшей обработки, определяет незадействованные каналы из перечня доступных для организации сети. Управляет сетевыми подчиненными устройствами, устанавливает параметры сети, число сетевых маршрутизаторов и число подчиненных устройств. Большую часть времени находится в режиме приема, обеспечивает организацию таблиц маршрутизации, позволяет маршрутизаторам и конечным устройствам входить в сеть. Обычно в сети один координатор.

2. ● Транзитные узлы или устройства маршрутизаторы (FFD-Full Function Device — устройство с полным набором функций), выполняют только передачу информации и управление маршрутизацией определяющей направления передачи. Маршрутизатор расширяет область покрытия сети, осуществляя динамическую маршрутизацию в обход препятствий, восстанавливает маршруты в случаях перегрузки сети или отказа какого-либо устройства. Маршрутизаторы работают в непрерывном режиме, имеют стационарное питание

3. ● Обычные узлы или конечные устройства (RFD - Reduced Function Device — устройство с ограниченным набором функций), осуществляют сбор информации в некоторой окрестности точки размещения данного узла. Могут принимать и отправлять пакеты, но не могут осуществлять их трансляцию. Конечные устройства подключаются к координатору или маршрутизатору. Конечные устройства могут переводиться в спящий режим для экономии заряда аккумуляторов. Именно конечные устройства взаимодействуют с датчиками, локальными контроллерами или исполнительными механизмами, их количество диктуется потребностью конкретного приложения. Сенсорные (конечные) узлы могут закрепляться стационарно, а также иметь относительную мобильность, то есть произвольно перемещаться друг относительно друга в некотором пространстве, не нарушая при этом логической связанности сети. В последнем случае сенсорная сеть не имеет фиксированной постоянной топологии, и ее структура динамически меняется с течением времени.

Сети DigiMesh

Особенностью протокола *DigiMesh* является возможность построения *mesh*-сети (см. рис.4) со спящими маршрутизаторами. В сети *DigiMesh* нет координатора с выделенной ролью — каждый из узлов сети может взять его функции на себя.

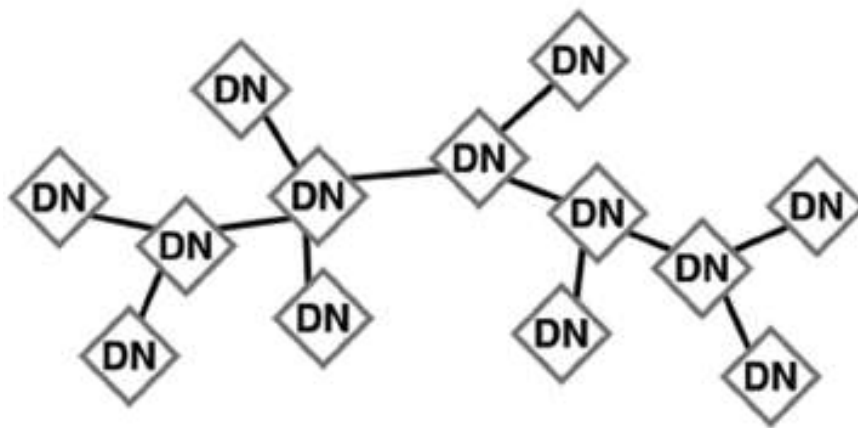


Рис.4. Пример DigiMesh сети

Возможность режима спящих маршрутизаторов в *DigiMesh* позволяет всем узлам в сети синхронизировать время работы и сна. Все синхронизированные узлы входят и выходят из режима низкого энергопотребления в одно и то же время. Так формируется циклически засыпающая сеть. Узлы синхронизируются путем получения специального *RF*-пакета (сообщения синхронизации), который отправляется узлом, играющим роль координатора сна. Узел сети может стать координатором сна, пройдя через процесс номинирования. Координатор сна отправляет одно сообщение синхронизации в начале каждого периода работы сенсорной сети. Сообщение синхронизации отправляется путем широковещательной рассылки. Время работы и сна всей сети в целом могут быть изменены локальными изменениями настроек на отдельных узлах сети.

Сети 6LoWPAN

В настоящее время для сопряжения БСС с сетями связи общего пользования (ССОП) применяется протокол *6LoWPAN* («*IPv6 over Low power Wireless Personal Area Networks*» – стандарт взаимодействия по протоколу *IPv6* поверх маломощных беспроводных персональных сетей стандарта *IEEE 802.15.4*), который позволяет интегрировать сенсорные сети в существующее семейство сетей.

Данный протокол позволяет передавать IP-пакеты поверх стандарта *IEEE 802.15.4* способом, удовлетворяющим открытым стандартам (протокол *IPv6*). При этом обеспечивается взаимодействие с другими IP-каналами и устройствами. Протокол *6LoWPAN* создан для маломощных беспроводных персональных сетей (*LoWPANs*) и описан в документах [RFC4919] и [RFC4944]. В архитектуре сети *6LoWPAN* (см. рис.5) определены три типа устройств (оконечный узел, маршрутизатор и шлюз), а также три вида сетей: «*Простая LoWPAN*», «*Расширенная LoWPAN*» и «*Ad hoc LoWPAN*». Здесь «*Ad hoc LoWPAN*» не подключена к ССОП, «*Простая LoWPAN*» подключена к ССОП через один шлюз, а «*Расширенная LoWPAN*» включает в себя несколько шлюзов, связанных с ССОП и друг с другом посредством магистральной линии связи.

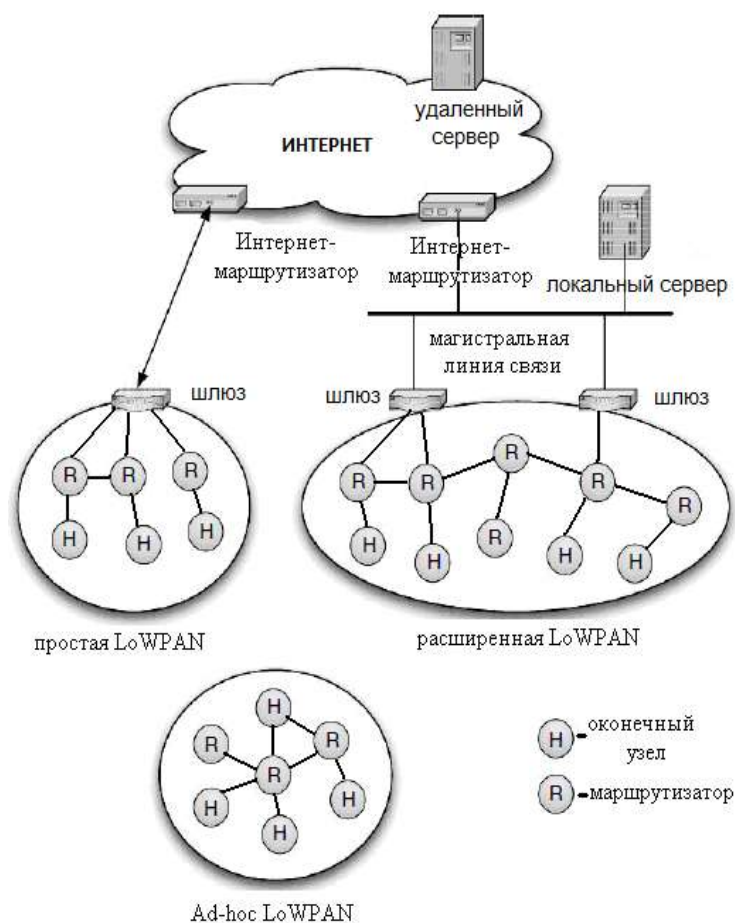


Рис.5. Архитектура сети 6LoWPAN

Здесь к названию технологии сенсорных сетей прибавляется выражение **ad hoc**, что означает «для данной цели», «по ситуации», то есть, датчики организовываются в «ad hoc» - сети для выполнения конкретных приложений.

Особенности применения БСС на лесных территориях

Ландшафтные особенности, географические характеристики лесных территорий и антропогенные воздействия на леса выдвигают на первый план решение задачи проектирования оптимального расположения узлов сети на лесных участках в выбранной архитектуре БСС. Леса с их неоднородностью строения, географического положения, плотности прорастания существенно влияют на возможности инфокоммуникационного взаимодействия внутри сетей. В работе [9] подробно изложены возможности использования БСС в лесах. Здесь отмечено, что путь прохождения радиоволн от маршрутизатора к приемнику датчика и обратно крайне разнообразен - от прямой видимости до направления сильно закрытого препятствиями, стволами и кронами деревьев. В отличие от проводной связи, где параметры постоянны, в беспроводной сети радиоканалы передачи данных имеют случайные параметры, часто сложно анализируемые. Механизм распространения радиоволн различен, но в основном может быть представлен отражением, дифракцией и рассеянием. Большинство систем рабо-

тают в местах с отражающими элементами, где нет прямой видимости антенн передатчика и приемника, а наличие высоких деревьев вызывает большие дифракционные потери. Благодаря многократным переотражениям от различных объектов, радиоволны проходят различный путь.

Моделирование распространения радиоволн, основано на предсказании среднего уровня принимаемого сигнала, на расстоянии от излучателя, а также определении разброса его значений в зависимости от конкретной ситуации на радиотрассе. Расчет радиоканала позволяет определить зону обслуживания узла опроса и конечного узла, т.е. места их расположения.

Рассмотренные особенности оказывают существенное влияние на решение задач

обоснования и выбора архитектуры сенсорной сети оптимального размещения узлов-сенсоров на больших территориях.

Помимо этих особенностей, влияющих на расстановку датчиков, следует фиксировать следующие факторы на лесной территории:

- наличие железной дороги;
- близость линий электропередач (ЛЭП);
- наличие автомобильных трасс, лесных и лесовозных дорог;
- удаленность технологических объектов от лесных территорий;
- наличие объектов охраны и защиты лесов;
- наличие просек и лесоустроительных знаков.

Учет перечисленных факторов необходим при реализации проекта БСС, так как это наиболее вероятные места возникновения лесных пожаров и желательно именно здесь или на близком расстоянии от этих объектов располагать сенсоры (конечные узлы).

Проектирование оптимальной расстановки датчиков обычно связано с разработкой и исследованием математических моделей, среди которых:

- графовые модели, учитывающие топологию местности, особенности функционирования сенсоров и реализации протокола ZigBee, а также выбранных технологий дальней радиосвязи;

- модели вычислительной геометрии - диаграммы Вороного («разбиение Вороного») как метод решения задачи поиска ближайшего соседа и учета перечисленных объектов [10]. Применительно к мониторингу с учетом ограниченной зоны действия датчиков принцип этого метода можно трактовать так, что зона контроля пространства каждого датчика должна распространяться до тех пор, пока она не коснется зоны действия соседнего.

В работе [11] представлены результаты исследований по определению влияния лиственного лесного массива на дальность связи в БСС. Здесь показано, что эффективность механизма маршрутизации и качества связи в сетях ZigBee зависит как от конфигурации самой сети (топологии и возможности самоорганизации) и параметров окружающей среды, так и от особенностей качества связи на физическом уровне сети и характеристик

приемопередатчиков. Проведенные исследования свидетельствуют о возможности создания в лесу распределенной ZigBee сети с межузловыми расстояниями от 60 до 120 метров.

Проблемой здесь является передача сигнала. Поскольку датчик реагирует на состояние достаточно ограниченного пространства, число необходимых передатчиков для хорошего функционирования системы велико, а лесной массив – далеко не самая благоприятная среда для распространения сигнала. Возникнет необходимость использования ретрансляторов, установленных на вышках (где лесной массив не мешает распространению сигнала).

На базе БСС в дальнейшем можно строить систему обнаружения, включающую в свой состав систему датчиков различного уровня, распределенных в окружающей среде с учетом рельефа местности и вероятности образования очагов возгорания; радиолокационные устройства, калиброванные для обнаружения дыма; тепловизоры и камеры; подсистему передачи данных, построенную на протоколах и оборудовании с низким электропотреблением; ГИС сервера; интернет ресурсы и web – технологии [12].

К одному из таких интернет ресурсов можно отнести – MapServer [13], как многофункциональную платформу для быстрого, легкого и эффективного распространения картографической информации, основанная на Map Guide Open Source. Возможностями MapServer являются: доступ к картографическим и проектным данным через Интернет, интуитивно понятные интерактивные карты, простота установки и развертывания, удобство формирования карт и создания картографических Web - приложений, интеграция и возможности взаимодействия с корпоративными системами управления базами данных (СУБД), мощные и гибкие возможности API-интерфейса. Веб-ориентированная геоинформационная система послойно отображает прогнозную информацию на электронной карте и использует подсистему управления базами данных (СУБД) для хранения лесотаксационных характеристик и описания различных объектов и населенных пунктов на контролируемой лесопокрытой территории.

Геоинформационная система, как информационно – аналитическая платформа содержит хранилище данных обо всех особенностях территорий покрытых лесами. Инструментально – это совокупность различных геообъектов геометрически представленных в виде множества точек той или иной плотности с измеряемыми расстояниями между ними. Данные, необходимые для функционирования систем мониторинга, есть - геоданные, характеризующие пространственное положение контролируемых объектов, а с этими данными лучше всего работать в среде ГИС.

В этой среде с использованием информационных web - технологий разработаны специализированные решения для мониторинга лесов. Так сейчас доступны ресурсы и инструменты, позволяющие без программирования создавать геоинформационные веб-приложения, связывающие во едином информационном пространстве с различных устройств и отображающие ее на инте-

рактивной электронной карте. При необходимости за счет специализированной разработки функционал таких приложений может быть расширен для создания более сложной аналитической системы.

Одна из классических функциональных возможностей геоинформационных систем - построение оптимального маршрута к месту лесного пожара, определение близлежащих сил и средства и построение для них персональных маршрутов для прибытия на место пожара [14].

Веб-приложения дополнительно обеспечивают просмотр большого числа специализированных картографических слоев, накладываемых на базовую карту, к примеру, таких как:

- вероятности возникновения лесных пожаров по конкретной причине антропогенного характера (из-за грозовой активности или по неустановленным причинам и т.п.);
- прогнозируемое число лесных пожаров на контролируемой территории с детализацией – лесничество;
- расположение территориальных органов управления, средств оповещения;
- территориальное противопожарное планирование и др.

Выполнение веб-приложения может осуществляться на любом компьютере или мобильном устройстве, подключенных к Интернет [15].

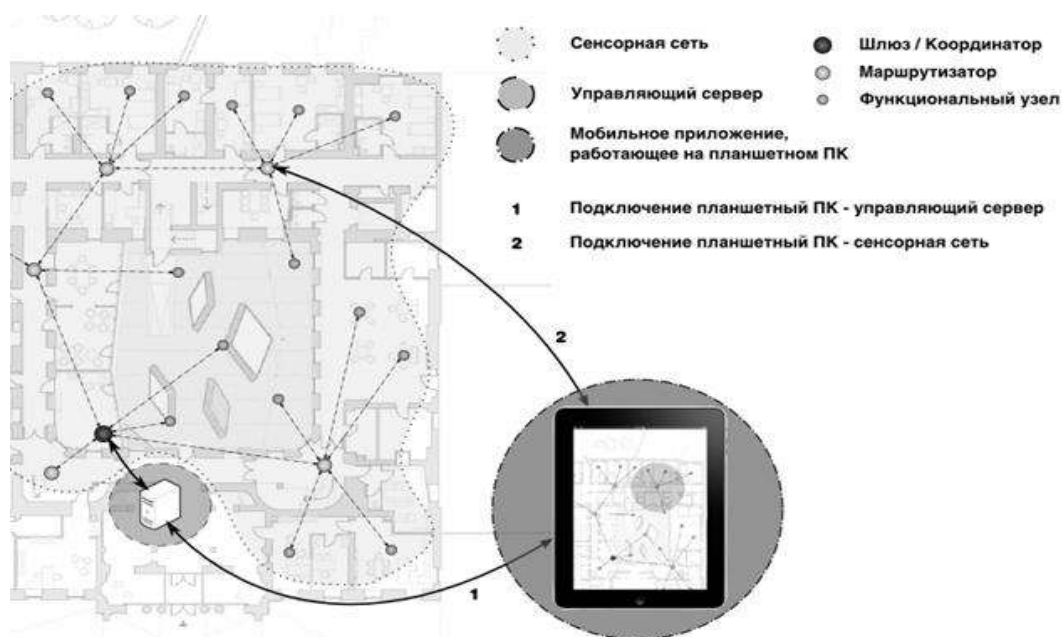


Рис. 6. Доступ к БСС с помощью планшетного ПК

На рисунке 6 представлена упрощенная схема доступа к беспроводной сенсорной сети, просмотра ее конфигурации, результатов измерений и контроля параметров окружающей среды с помощью планшетного ПК.

Здесь показаны следующие компоненты: сенсорная сеть, управляющий сервер, мобильное web - приложение, работающее на планшетном ПК.

Управляющий сервер выполняет программу, которая взаимодействует со шлюзом БСС и осуществляет мониторинг и управление работой сенсорной сети.

На планшетном ПК мобильного пользователя установлено приложение, которое может безопасно подключаться к управляющему серверу как через встроенные в планшетный ПК технологии передачи данных (Wi-Fi, 3G, Bluetooth) так и через сенсорную сеть с помощью трансивера, работающего по тому же протоколу, что и сенсорная сеть.

При подтверждении прогнозов распространения сенсорных сетей и планшетных ПК можно ожидать появления планшетных ПК со встроенными трансиверами. Внешний трансивер уже сейчас может быть подключен к планшетному ПК либо с помощью кабеля, либо по стандарту беспроводной связи, такому как Bluetooth. В этом случае, перемещаясь по территории, на которой развернута сенсорная сеть, планшетный ПК пользователя может устанавливать до 2 сетевых подключений: между планшетным ПК и управляющим сервером; между трансивером, подключенным к планшетному ПК, и сенсорной сетью.

Первое подключение может использоваться, например, для загрузки плана территории, на которой развернута БСС, получения информации о координатах всех узлов и передачи пользователем управляющих запросов к серверу и отдельным узлам сети.

По второму подключению трансивер осуществляет взаимодействие с сенсорной сетью и передает как мобильному приложению, так и управляющему серверу данные об узлах БСС в пределах собственной радиовидимости. Подобный механизм обратной связи при необходимости позволит информировать мобильных пользователей о местоположении друг друга и осуществлять обмен информацией между ними.

Заключение

Беспроводные сенсорные сети являются одним из самых быстро развивающихся направлений современных телекоммуникационных технологий. Перспективы их использования связаны как с заменой кабельной инфраструктуры на радиоэфир, так и с новыми функциональными возможностями и областями применения.

Благодаря таким характеристикам беспроводных сенсорных сетей, как миниатюрность узлов, низкое энергопотребление, встроенный радиointерфейс, достаточная вычислительная мощность, сравнительно невысокая стоимость, стало возможным их широкое использование во многих сферах человеческой деятельности с целью автоматизации процессов сбора информации, мониторинга и контроля характеристик разнообразных технических и природных объектов.

Библиографический список

1. Профилактика и меры предупреждения лесных пожаров в системе лесоправления Российской Федерации / Д. Ф. Ефремов, А. С. Захаренков, М. А. Копейкин, Е. П. Кузьмичев, М. И. Сметанина, В. В. Солдатов ; под общ ред. Е. П. Кузьмичева. – М.: Всемирный банк, 2012. – 104 с
2. Кузнецов Г.В., Барановский Н.В. Прогноз возникновения лесных пожаров и их экологических последствий. Новосибирск: Изд-во СО РАН, 2009.
3. Шерстюков Б.Г. Методы оценки последствий изменения климата для физических и биологических систем: коллективная монография. – М.: Росгидромет, 2012.
4. Е.В. Воробьев, А.Я. Номерчук, К.К. Арутюнов, Г.Г. Лисовой, В.И. Плотников Технические компоненты для создания системы раннего обнаружения лесных пожаров. Известия ЮФУ. Технические науки №8 (169), 2015.
5. Плахтеев А.П., Орехов А.А., Плахтеев П.А. Система мониторинга лесных пожаров на основе сенсорных сетей // Радіоелектронні і комп'ютерні системи. Науково – технічний журнал – 2014.
6. Научно-информационный материал «Современные и перспективные технологии построения беспроводных сенсорных сетей» <http://www.srdmtuci.ru/attachments/article/79/pr%202.4.DOC>
7. Мочалов В.А., Турута Е.Н. Стратегии размещения узлов сенсорной сети // Материалы VII Международной научно-технической конференции. М. 2010.
8. С.П.Хабаров. Компьютерные сети. <http://www.habarov.spb.ru/net/index.htm>
9. Отчет о НИР «Экологический и экспериментальный мониторинг лесов и управление ими на основе RFID-технологии». ГОУ ВПО «Уральский государственный лесотехнический университет». 2009.
10. Васильев Н.П., Заяц А.М., Логачев А.А. Реализация алгоритма Форчуна расчета диаграмм Вороного на PHP. Свидетельство о государственной регистрации программ для ЭВМ № 2015616364 от 08.06. 2015.
11. Галкин П.В., Головкина Л.В. Исследование влияния лесных массивов на дальность связи в сетях ZigBee. Восточно-Европейский журнал передовых технологий. 3/2 (51). 2011.
12. Создание картографических сервисов с использованием MapServer. <http://gis-lab.info/qa/mapserver.html>
13. Барановский Н.В., Жарикова М.В., Ляшенко Е.Н. Концептуальный проект веб-ориентированной географической информационной системы прогноза лесной пожарной опасности. <http://conf.nsc.ru/intercarto17/reportview/88910>.
14. Заяц А.М., Логачев А.А. Мультимедийное web-приложение мониторинга, прогнозирования и принятия решения по тушению лесных пожаров.

Свидетельство о государственной регистрации программ для ЭВМ № 2012618909 от 02.10.2012

15. Мочалов В.А., Турута Е.Н., Халецкий А.А. Навигация и мониторинг в сенсорной сети с использованием планшетных ПК. Материалы IX Международной НТК «Перспективные технологии в средствах передачи информации – ПТСПИ-2011». — Владимир-Суздаль, 2011.

И.В.Панфилов, доктор технических наук, профессор

ИНФОРМАЦИОННАЯ ТЕХНОЛОГИЯ ЗАЩИТЫ ИНФОРМАЦИИ В КОМПЬЮТЕРНЫХ СИСТЕМАХ

Технико-экономическое обоснование эффективности системы защиты

Задачей технико-экономического обоснования (ТЭО) варианта оборудования объекта техническими средствами защиты является выбор рационального варианта, который бы удовлетворял минимуму капитальных вложений собственников объекта в систему защиты.

Рассмотрим выбор оптимального варианта на этапе проектирования по критерию экономической эффективности системы $-\mathcal{E}_c$:

$$\mathcal{E}_c = (K - K_{обс}) / A_{cob} \rightarrow \max,$$

где K – потенциальный выигрыш в применении системы защиты по сравнению с вариантом отсутствия защиты объекта;

$$K = P_{ny} \cdot C_{мц} - [(1 - P_{ny}) \cdot A_{зм} + A_{кан}],$$

где P_{ny} – вероятность потенциальной угрозы кражи в течении года на объекте; $C_{мц}$ – стоимость материальных ценностей на объекте; $A_{зм}$ – годовой тариф оплаты за охрану объекта; $A_{кан}$ – приведенная к одному году стоимость затрат собственника на капитальные вложения в охрану объекта с учетом срока службы технических средств; A_{cob} – ежегодные затраты собственника на охрану объекта; $K_{обс}$ – дополнительные затраты на обслуживание объектового комплекса защиты:

$$K_{обс} = C_{рем} / T_{cpi} + m_{yk} \cdot C_y,$$

где $T_{cpi} = R_n \sum_{i=1}^n T_{cpi}$ – средняя наработка на отказ комплекса защиты; T_{cpi} –

средняя наработка на отказ i -го средства защиты; R_n – поправочный коэффициент, предназначенный для учета специфики условий эксплуатации средств защиты на конкретном объекте ($R_n = 1 \pm 0,15$); $C_{рем}$ – средняя стоимость

ремонта i -го средства; $m_{yk} = \sum_{i=1}^n m_{yi}$ – количество условных установок;

m_{yi} – количество условных установок i -го типа; C_y – стоимость проведения технического обслуживания одной условной установки.

Концептуальная модель охранной системы защиты КС

Систему защиты информации 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 информации 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 от 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 $x_i R x_j$, то от вершины x_i проводят ориентированную дугу к вершине x_j ; если $x_i \bar{R} x_j$, то дуга отсутствует [1].

Пусть элементами системы охранной защиты информации 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 опознания 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,

A_{14} – выход из охраняемого объекта.

Тогда концептуальную модель охранной системы защиты информации можно представить в виде граф-схемы, основываясь на концептуальной модели системы безопасности КС, представленной на рисунке 1.

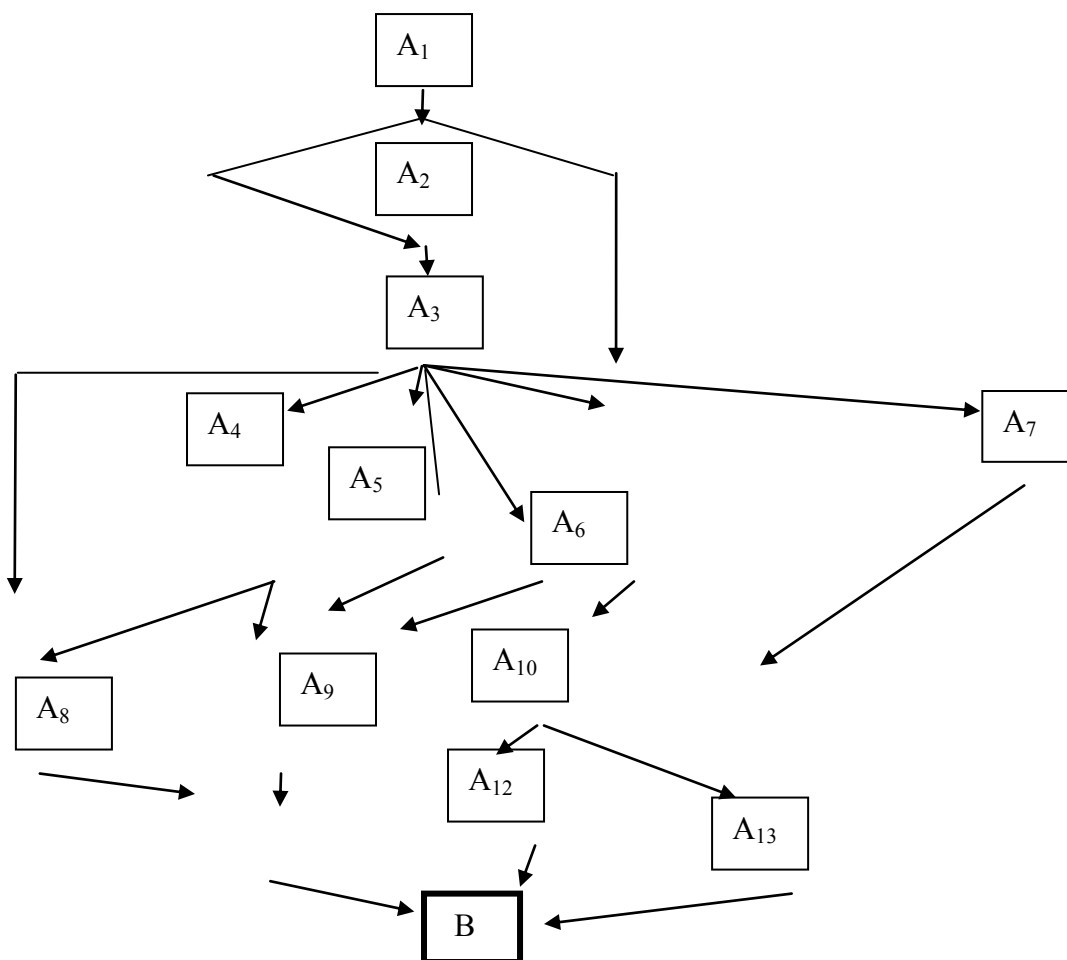


Рис.1. Граф-схема концептуальной модели системы защиты

Метод экономического расчета затрат на проектируемую охранную систему защиты КС

Этот метод осуществляется в два этапа:

- 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 (в рублях), который предприятие может понести 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 ($C_{\kappa 1}$) от утечки данной информации?
2. Каков денежный ущерб ($C_{\kappa 2}$) от утраченной выгоды предприятия?
3. Каков денежный ущерб ($C_{\kappa 3}$) от замедления дел предприятия?
4. Какие усилия будут необходимы для нейтрализации причиненного ущерба ($C_{\kappa 4}$)?

Возможный денежный ущерб (C_{κ}) от утечки или разрушения каждого κ -го сведения подсчитывается как сумма $C_{\kappa i}$:

$C_{\kappa} = C_{\kappa 1} + C_{\kappa 2} + C_{\kappa 3} + C_{\kappa 4}$, где $\kappa = 1, 2, \dots, n$ – число сведений, подлежащих защите.

В основу экономического расчета распределения затрат на проектируемую охранную систему защиты информации положена концептуальная модель (см. рис.5.2).

При анализе модели системы защиты информации предприятия необходимо использовать положения системного анализа, согласно которым декомпозиция этой системы представлена схемой концептуальной модели. Поведение этой системы можно определить вычислением поведения наибольшего элемента разложения, используя арифметико-логические связи [2]:

$$A_j = \sum_{i=1}^n a_{ij} \cdot A_i, \text{ где } A_i \text{ – вершина, а } a_{ij} \text{ – дуги графа}$$

Арифметико-логические связи нашей схемы (см. рис.5.2) имеют вид:

$$\begin{aligned} A_2 &= a_{12} \cdot A_1, & A_3 &= a_{23} \cdot A_2 + a_{13} \cdot A_1, & A_4 &= a_{34} \cdot A_3, \\ A_5 &= a_{35} \cdot A_3, & A_6 &= a_{36} \cdot A_3 + a_{16} \cdot A_1, & A_7 &= a_{37} \cdot A_3, \\ A_8 &= a_{38} \cdot A_3 + a_{48} \cdot A_4, & A_9 &= a_{49} \cdot A_4 + a_{59} \cdot A_5 + a_{39} \cdot A_3, & A_{10} &= a_{310} \cdot A_3 + a_{610} \cdot A_6, \\ A_{11} &= a_{911} \cdot A_9 + a_{811} \cdot A_8, & A_{12} &= a_{1012} \cdot A_{10}, \\ A_{13} &= a_{1013} \cdot A_{10} + a_{713} \cdot A_7, & A_{14} &= a_{1114} \cdot A_{11} + a_{1214} \cdot A_{12} + a_{1314} \cdot A_{13}. \end{aligned}$$

Решая приведенные логические уравнения, находим пути достижения вершины A_{14} . Результаты вычисления сведены в матрицу смежности $Q = [q_{ij}]_{m \cdot n}$ (табл. 5.1), каждому элементу которой взаимно однозначно соответствует элемент системы защиты, входящий в событие (путь), каждой строке S_{cs} – 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 $q_{ij} = \begin{cases} 1, & \text{если } j\text{-й элемент входит в } i\text{-ю совокупность элементов,} \\ & \text{при наличии которых событие истинно,} \\ 0, & \text{в противном случае.} \end{cases}$

Таблица 1

Матрица смежности Q

$S \backslash A$	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}	A_{14}
S_1	1	1	1					1			1			1
S_2	1		1					1			1			1
S_3	1		1			1					1			1
S_4	1		1	1				1			1			1
S_5	1	1	1	1				1			1			1
S_6	1	1	1	1					1		1			1
S_7	1		1	1					1		1			1
S_8	1	1	1		1				1		1			1
S_9	1		1		1				1		1			1
S_{10}	1	1	1								1			1
S_{11}	1		1								1			1
S_{12}	1	1	1							1		1		1
S_{13}	1	1								1		1		1
S_{14}	1	1	1			1				1		1		1
S_{15}	1		1			1				1		1		1
S_{16}	1					1				1		1		1
S_{17}	1	1	1							1			1	1
S_{18}	1		1							1			1	1
S_{19}	1	1	1			1				1			1	1
S_{20}	1		1			1				1			1	1
S_{21}	1					1				1			1	1
S_{22}	1	1	1				1						1	1
S_{23}	1		1				1						1	1

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}	A_{14}
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
A_{14}	23	11	20	4	2	7	2	4	4	10	11	5	7	23

Таблица 3

Транспонированная матрица Q^T

S A	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
A1	1	1	1	1	1	1	1	1	1	1	1	1	1
A2	1				1	1		1		1		1	1
A3	1	1	1	1	1	1	1	1	1	1	1	1	
A4				1	1	1	1						
A5								1	1				
A6			1										
A7													
A8	1	1		1	1								
A9						1	1	1	1				
A10												1	1
A11	1	1	1	1	1	1	1	1	1	1	1		
A12												1	1
A13													
A14	1	1	1	1	1	1	1	1	1	1	1	1	1

Таблица 4

Таблица весов направленных дуг

	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	B
A1														
A2	0,1													
A3	0,00 5	0,0 8												
A4	0,45	0,5 2	0,33											
A5	1	0,4 3	0,76	0										
A6	0,28	0,3 8	0,4	0	0									
A7	1	0,4 3	0,76	0	0	0								
A8	0,45	0,5 2	0,33	0,19	0	0	0							
A9	0,45	0,2 6	0,33	0,19	0,09	0	0	0						
A10	0,12	0,2 1	0,26	0	0	0,08	0	0	0					
A11	0,46	0,2 3	0,06	0,16	0,43	0,67	0	0,17	0,17 0	0				
A12	0,34	0,3 2	0,85	0	0	0,19	0	0	0	0,09	0			
A13	0,22	0,3 2	0,28	0	0	0,26	0,23	0	0	0,13	0	0		
B	0	0,1	0,02 8	0,45	1	0,22	1	0,45	0,4	0,98	0,7	0,34	0,27	

Таблица оценки прочности путей

S_i	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}	S_{10}	S_{11}
P_{Si}	0,58	0,5	0,85	0,66	0,62	0,66	0,65	0,86	0,84	0,29	0,29	0,22

S_i	S_{12}	S_{13}	S_{14}	S_{15}	S_{16}	S_{17}	S_{18}	S_{19}	S_{20}	S_{21}	S_{22}	S_{23}
P_{Si}	0,64	0,59	0,7	0,69	0,6	0,61	0,57	0,71	0,68	0,58	0,89	0,87

Для определения узких мест в системе защиты, т.е. поиск элементов защиты, находящихся под наибольшей угрозой, необходимо определить интенсивность участия элементов защиты в путях несанкционированного доступа к информации. Интенсивность участия элементов в событии можно найти, используя матрицу отношений $F = (w_{ij})_{n \cdot n}$ [3].

Частотная матрица отношений F , матрица смежности которой Q имеет вид (табл. 2):

$$F = Q^T \cdot Q = \begin{vmatrix} w_{11} & w_{12} & w_{13} & \dots & w_{1n} \\ w_{21} & w_{22} & w_{23} & \dots & w_{2n} \\ \dots & \dots & \dots & \dots & \dots \\ w_{n1} & w_{n2} & w_{n3} & \dots & w_{nn} \end{vmatrix}$$

где Q^T – матрица, транспонированная по отношению к Q (табл. 5.3).

Для оценки путей S_i граф-схемы концептуальной модели вычисляются веса направленных дуг (табл. 4).

Ущерб, причиненный компьютерными преступлениями, оценить 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

$$P_{si} = [1 - \prod_{\substack{i=1 \\ j=i+1}} (1 - p(a_{ij}))]$$

где $p(a_{ij})$ – нормированные значения весов (вероятностей) дуг граф-схемы концептуальной модели.

Рассчитанные вероятности P_{si} (табл. 5.5) позволят более объективно спроектировать и организовать систему защиты информации. Для расчета распределения затрат на проектируемую охранную систему защиты используем метод максимального элемента [4].

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 C_3 условных единиц для повышения прочности отдельных элементов так, чтобы суммарная прочность системы защиты была наибольшей.

Рассмотрим распределение затрат на элементы наиболее прочного пути $S_{кр}$.

Пусть $S_{кр}$ состоит из 5 элементов, соединенных последовательно. Значения вероятности (преодоления прочности, которую задаем в начале проектирования системы) P_i и стоимости C_i элементов даны в табл. 5.6. Требуется зарезервировать систему так, чтобы ее суммарная стоимость не превышала C_3 , а прочность была бы максимальной.

Принимаем $C_3 = C_0 + C_{дон}$

Т а б л и ц а 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 будет

$$C_0 = \sum C_i = 5,6 \text{ у.е.}$$

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. 5.7.

Прирост прочности элементов пути на 1 у.е.

	$k \setminus i$	1	2	3	4	$N=5$
$\alpha(k, i)$	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^0		2	2	5	4	4

Выбирая из табл.5.7 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:

$$(C_{don}) = 0,4 \rightarrow 0,5 \rightarrow 3 \rightarrow 0,4 \rightarrow 0,5 \rightarrow 0,7 \rightarrow 1 \rightarrow 3 \rightarrow 3 \rightarrow 0,4 \rightarrow 0,5 \rightarrow 3 = 16,4$$

Полная стоимость элементов $S_{кр}$

$$C_3 = C_0 + C_{don} = 5,6 + 16,4 = 22 \text{ у.е.}$$

Расчет для остальных элементов системы защиты, которые входят в пути S_5, S_8 и S_{14} , за исключением элементов, входящих в $S_{кр}$, т.е. S_{22} , можно выполнить аналогично.

Библиографический список

1. Панфилов И.В., Заяц А.М., Панфилова Е.И. Проблема защиты информации в компьютерных системах – Учебное пособие. СПб ГЛТА, 2008, 148с.
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.

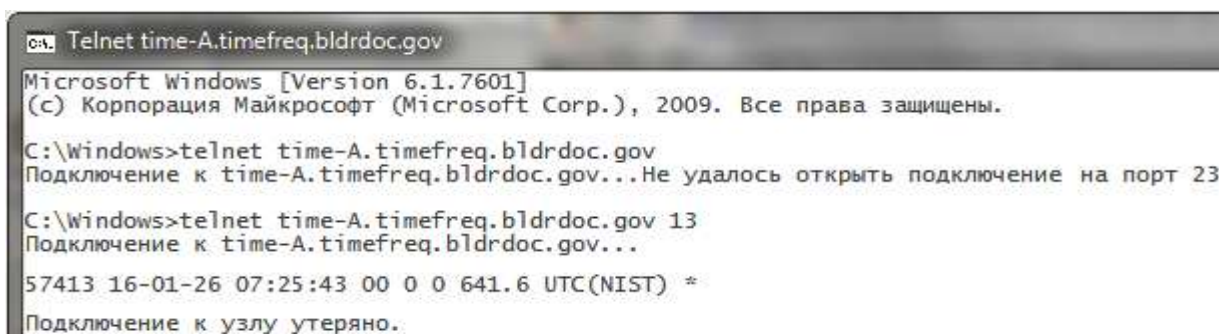
4 . Березин Е.А. Оптимальное распределение ресурсов и элементы синтеза систем / Под ред. Е.В. Зологова. – М.: Советское радио, 1974 – 304 с.

С.П. Хабаров, кандидат технических наук, доцент

ТЕРМИНАЛЬНЫЙ ДОСТУП МЕЖДУ UBUNTU И WINDOWS КОМПЬЮТЕРАМИ

Даже небольшая сеть может содержать несколько серверов, находящихся в разных подразделениях. Странно будет выглядеть администратор этой сети, бегущий между этажами для локализации какого-либо сбоя. А что ему делать, если надо на всех компьютерах обновить какой-либо софт? Ясно, что администрирование такой сети требует наличие удаленного доступа, как минимум к основным серверам и как максимум ко всем компьютерам ЛВС.

Исторически первым для удаленного доступа использовался протокол telnet. Во всех сетевых операционных системах (ОС) есть свой telnet-клиент, который отображает консоль удаленного компьютера, что позволяет вводить команды и получать результат (рис. 1).



```
cmd. Telnet time-A.timefreq.bldrdoc.gov
Microsoft Windows [Version 6.1.7601]
(c) Корпорация Майкрософт (Microsoft Corp.), 2009. Все права защищены.

C:\Windows>telnet time-A.timefreq.bldrdoc.gov
Подключение к time-A.timefreq.bldrdoc.gov...Не удалось открыть подключение на порт 23

C:\Windows>telnet time-A.timefreq.bldrdoc.gov 13
Подключение к time-A.timefreq.bldrdoc.gov...
57413 16-01-26 07:25:43 00 0 0 641.6 UTC(NIST) *
Подключение к узлу утеряно.
```

Рис. 1. Telnet-доступ к одному из серверов мирового времени

Со временем telnet устарел, и ему на смену пришел протокол SSH (Secure Shell), который отличается от telnet тем, что все данные, включая пароли доступа к удаленному компьютеру, передаются в зашифрованном виде.

Использование SSH — это легкий и безопасный способ получить доступ к удаленному компьютеру с ОС Ubuntu. Для установки на Ubuntu-компьютере SSH-сервера надо в консоли ввести команду:

```
sudo apt-get install openssh-server
```

При наличии доступа в Интернет установка выполнится автоматически, при отсутствии доступа надо иметь соответствующий deb-пакет. После установки SSH-сервера становятся доступными и функции SSH-

клиента. Для тестирования можно выполнить клиентский запрос к базовому компьютеру, который выступит в качестве удаленного узла, используя команду:

```
ssh localhost
```

Если SSH-сервер работает, то после ввода пароля можно вводить любые консольные команды, доступные на удаленном узле. Для завершения SSH-соединения используют команда `exit`. По умолчанию протокол SSH работает на 22 порту, но его можно и изменить. Проверить, что этот порт доступен извне можно, если использовать Система -> Администрирование -> Сетевые инструменты и задать на вкладке «Сканирование портов» IP-адрес интересующего вас Ubuntu-компьютера (рис. 2).

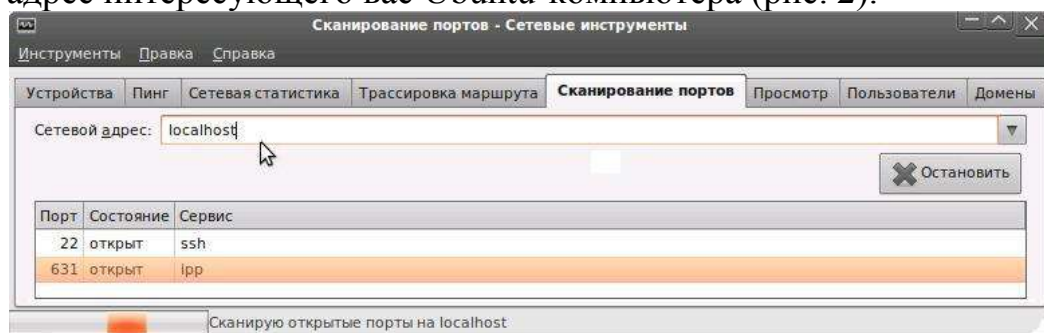


Рис. 2. Тестирование открытых портов на компьютере localhost

Для запуска, остановки и перезапуска SSH-сервера используют команды:

```
sudo /etc/init.d/ssh [start ИЛИ stop ИЛИ restart]
```

Все настройки SSH-сервера хранятся в файле `/etc/ssh/sshd_config`, а настройки клиента в файле `/etc/ssh/ssh_config`. Настройки клиента обычно не меняют, поскольку они приемлемы по умолчанию. Для сервера не помещает дополнительная настройка безопасности в файле `/etc/ssh/sshd_config`:

- *Port 22* – Задание номера порта, на котором работает SSH-сервер. Рекомендуется изменить.
- *PermitRootLogin no* – Запрет подключения к SSH-серверу, используя логин суперпользователя.
- *PermitEmptyPasswords no* – Запрет подключения пользователей, у которых пустые пароли. Рекомендуется, даже если вы единственный пользователь в системе
- *AllowUsers ...* – Разрешение на подключения только указанных пользователей. Логин пользователей разделяются пробелом.

После изменений файла конфигурации необходимо перезапустить SSH-сервер и убедиться в его работоспособности. А вот как и чем подключаться к SSH-серверам? Вопрос не праздный, так как это зависит от среды, из которой осуществляется подключение.

В системах на базе Linux, в том числе и в ОС Ubuntu, для подключения к удаленному компьютеру используют команду:

```
ssh [опции] <логин>@<адрес_удаленного_компьютера>
```

В качестве адреса можно использовать как IP-адрес, так и доменное имя компьютера. Наиболее часто используются следующие опции:

- c Задаёт список шифров, в порядке предпочтения через запятые.
- f Переводит ssh в фоновый режим после аутентификации.
- p Определяет порт SSH-сервера (по умолчанию 22).
- i Указывается пользователь, от имени которого нужно зарегистрироваться на удаленном компьютере.

Простейший формат команды подключения к удаленному SSH-серверу имеет вид:

```
ssh serp@vmUbuntu
```

где serp — имя пользователя на удаленной системе, а vmUbuntu — имя удаленного узла, вместо которого можно указать его IP-адрес.

Если SSH-сервер использует порт, отличный от стандартного, например, 1010, то команда на удаленный доступ будет иметь вид:

```
ssh -p 1010 user@hostname
```

Для удаленного администрирования Ubuntu-компьютеров из среды ОС Windows обычно используют программу PuTTY. Существует множество и других программ, но PuTTY — это популярный бесплатный SSH-клиент для безопасного подключения к удаленному компьютеру. Для начала работы с ней достаточно запустить файл PuTTY.exe (рис. 3).

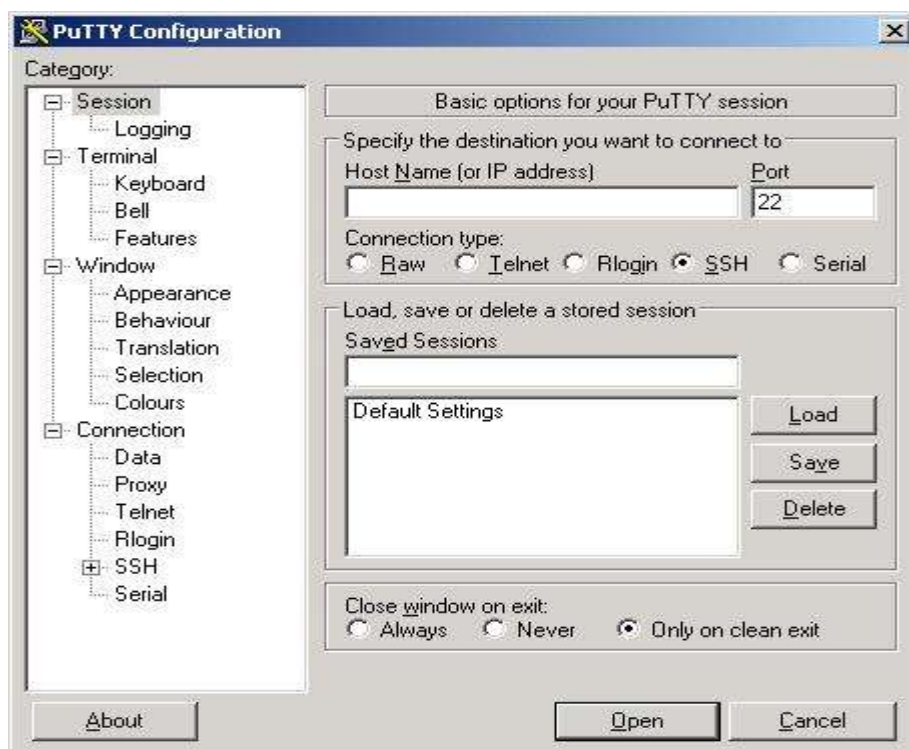


Рис. 3. Окно конфигурации утилиты PuTTY

Последовательность действий по конфигурированию PuTTY для подключения к удаленному SSH-серверу будет следующей:

- В поле «Host Name (or IP address)» вводят имя или IP-адрес удаленного хоста, с которым надо соединиться по SSH-протоколу.
- Порт по умолчанию 22 или тот, на который настроен SSH-сервер.
- В поле «Saved Sessions» вводят имя сессии и для сохранения ее параметров нажимают кнопку Save. Это позволит в дальнейшем из сохраненных сессий загружать нужную сессию (кнопка «Load»), не вводя заново требуемые параметры.
- Для связи с SSH-сервером надо нажать кнопку Open и появится окно (рис. 4), только абсолютно пустое, с единственной строкой сверху.



```
serp@vmUbuntu10: ~
login as: serp
serp@192.168.1.10's password:
Linux vmUbuntu10 2.6.32-24-generic #39-Ubuntu SMP Wed Jul 28 06:0
686 GNU/Linux
Ubuntu 10.04.1 LTS
Welcome to Ubuntu!
 * Documentation:  https://help.ubuntu.com/

Your CPU appears to be lacking expected security protections.
Please check your BIOS settings, or for more information, run:
  /usr/bin/check-bios-nx --verbose

Last login: Sat Jul  9 21:45:20 2011 from 192.168.1.2
serp@vmUbuntu10:~$ ls
examples.desktop  Share          Видео          Загрузки      Музыка
primer.txt        Ubuntu One    Документы     Картинки     Рабочий стол
serp@vmUbuntu10:~$
```

Рис. 4. Удаленная терминальная консоль узла 192.168.1.10

- В строке «login as» надо ввести логин доступа по SSH и нажать Enter. При успешной аутентификации потребуется ввести пароль доступа. Во время ввода пароля на экране ничего не отображается.
- Если логин и пароль введены правильно, то выполнится подключение к серверу, и Вы попадете в командную строку Ubuntu Linux.
- Далее можно использовать любые консольные команды, которые будут выполняться на удаленном сервере.

Если на удаленном компьютере установлен Midnight Commander, то в командной строке консоли следует выполнить команду mc. Это значительно упростит работу с удаленным Ubuntu-компьютером (рис. 5).

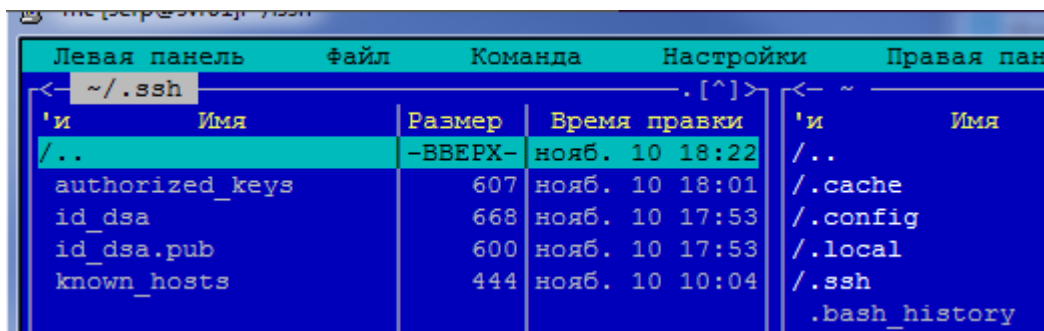


Рис. 5. Окно Midnight Commander удаленного компьютера

Однако, технологии развиваются, каналы связи становятся все более скоростными, и графический интерфейс преследует нас во всех сетевых устройствах, от мощных компьютеров до iPad'ов и iFon'ов.

Доступ из Ubuntu к удаленному рабочему столу Windows

Управлять Windows из Ubuntu можно подобно тому, как используется подключение к удаленному рабочему столу между платформами Microsoft. Для этого на машине с Windows. надо иметь разрешение на подключение к удаленному рабочему столу с вашей учетной записью.

Существующие приложения по работе с удаленными рабочими столами используют два различных типа протоколов:

- RDP (Remote Desktop Protocol) — протокол удаленного рабочего стола),
- VNC (Virtual Network Computing) — протокол виртуальной вычислительной сети.

Для возможности удаленного подключения оба узла, как сервер, так и клиент должны поддерживать один и тот же протокол. Проблема в том, что не все ОС используют по умолчанию одинаковые протоколы. Протокол RDP разработан корпорацией Microsoft для обеспечения доступа, контроля и управления ресурсами другого компьютера под управлением ОС Windows.

Для доступа из Ubuntu к Windows достаточно использовать консольную утилиту rdesktop, которая содержится в стандартной поставке любой версии Ubuntu. Работает она предельно просто, если известно имя или IP удаленного сервера RDP. Чтобы подсоединиться к удаленному рабочему столу Windows, надо в консоли Ubuntu ввести команду:

```
rdesktop [опции] <адрес_удаленного_компьютера>
```

Получить все доступные опции можно, запустив эту утилиту без параметров. Наиболее часто используемыми являются:

- u – Учётная запись для аутентификации.
- d – Домен.
- p – Пароль, если опция не используется, то в этом случае rdesktop запросит пароль при запуске.
- k – Эмулируемая раскладка клавиатуры.
- g – Разрешение экрана, можно указать в процентах от всего экрана.
- f – Полноэкранный режим (можно переключиться с помощью комбинации клавиш Ctrl-Alt-Enter).
- s – Начальная оболочка пользователя (вместо Explorer).
- c – Начальный рабочий каталог пользователя.
- 0 – Глубина цвета: 8, 16 или 24 бит. Основной для Ubuntu 24-битный цвет Windows XP/2003 не поддерживает.
- z – Активация сжатия передаваемых данных, актуально для медленных соединений.

Используя опции и флаги утилиты rdesktop, появляется возможность существенно модифицировать эту простую команду:

- Если необходимо указать определенную раскладку клавиатуры, то команда будет иметь вид — `rdesktop -k en_us 192.168.1.44`
- Указать размер открываемого окна в пикселях или в процентах можно командой — `rdesktop -g 60% -k en_us 192.168.1.44`
- При этом 100 % – это не полный экран, а полное окно, в котором вы работаете. Если нужен полный экран, то надо использовать следующую команду — `rdesktop -f -k en_us 192.168.1.44`
- Указать логин и пароль, чтобы не вводить их при установке соединения можно в команде — `rdesktop -u admin -p psw -k en_us 192.168.1.44`

Если активны две системы vmUbuntu и vmWinXP (192.168.1.44), а в последней установлен доступ к ее рабочему столу пользователю `serp` с паролем `serp`, то после ввода в терминале vmUbuntu команды:

```
rdesktop -g 60% -u serp -p serp 192.168.1.44
```

вид экрана базового компьютера будет иметь вид, приведенный на рис. 6.

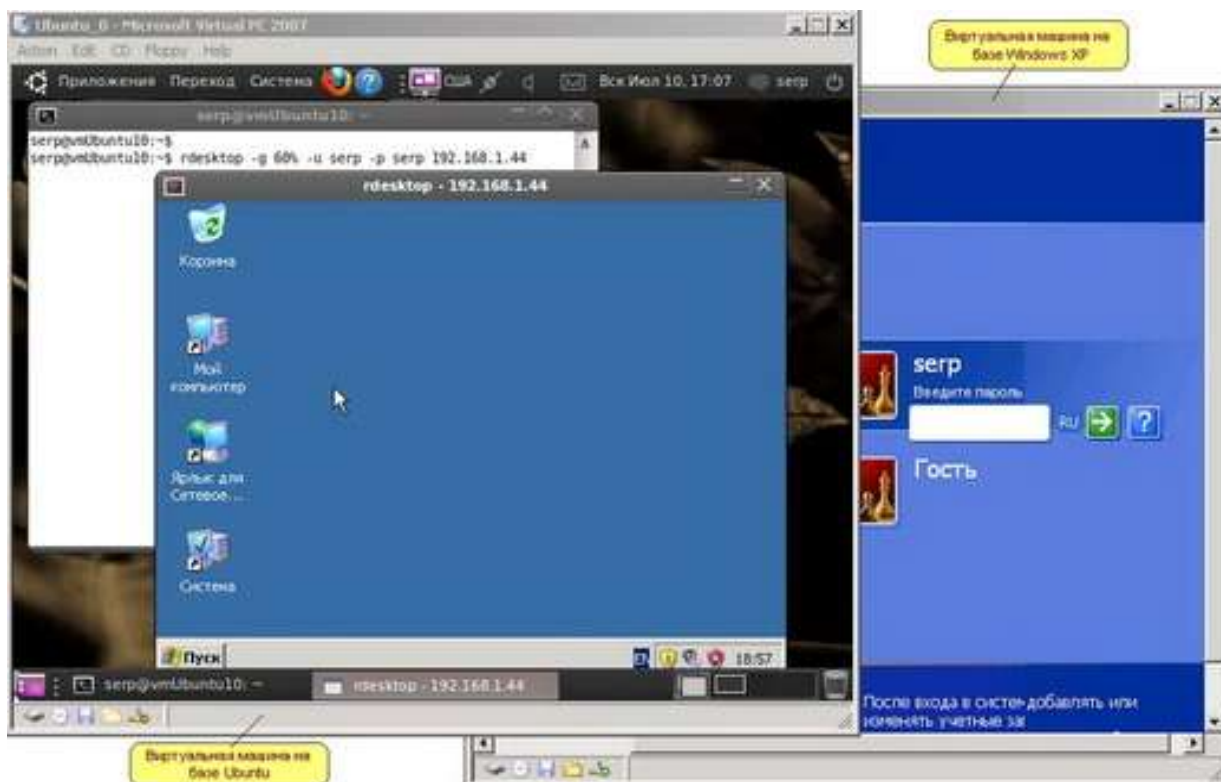


Рис. 6. Окна виртуальных машин основного ПК при доступе с Ubuntu-машины на Windows-машину

При этом `vmWinXP` — заблокируется, а в окне «`rdesktop - 192.168.1.44`» на `vmUbuntu`, будут доступны любые действия с Windows XP. Естественно, в рамках тех прав, которыми наделен пользователь ее удаленного рабочего стола.

Настройка доступа к удаленному рабочему столу в Ubuntu

Для этого в ОС Ubuntu можно использовать опцию основного меню Система -> Параметры -> Удаленный рабочий стол. Откроется окно, которое после выбора первой строки будет иметь вид, как на рис. 7.

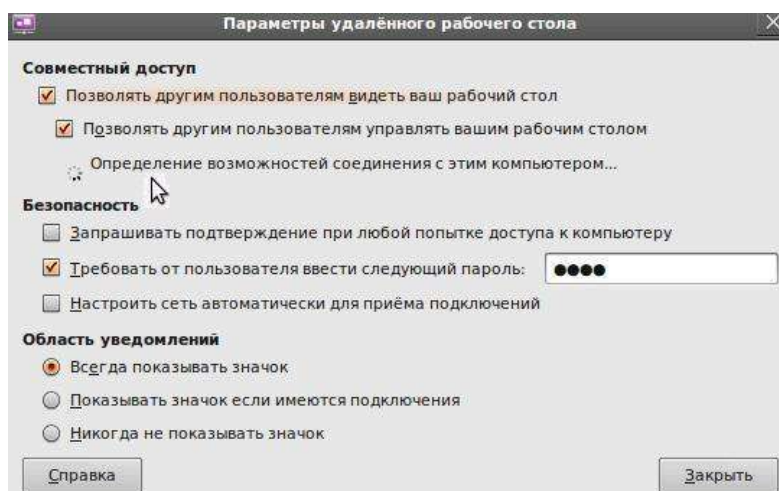


Рис. 7. Окно определения параметров подключения к рабочему столу Ubuntu

Надо обратить внимание на анимированную иконку в начале третьей строки. Идет процесс определения возможности работы этого компьютера в режиме удаленного рабочего стола. Если проверка пройдет штатно, то третья строка этого окна изменит свое содержание. В ней будет указан адрес, по которому можно обращаться извне к вашему рабочему столу (рис. 8).



Рис. 8. Окно параметры рабочего стола после проверки

Если на Windows-компьютере выполнить Все программы -> Стандартные -> Связь -> Подключение к удаленному рабочему столу, — то нас ожидает глубокое фиаско с отказом в доступе.

Это была связь Windows с Ubuntu, но если к этому же компьютеру vmUbuntu10 подключиться из другого Ubuntu-компьютера, используя Приложение -> Интернет -> Просмотр удаленных рабочих столов, то будет получен положительный результат (рис. 9).

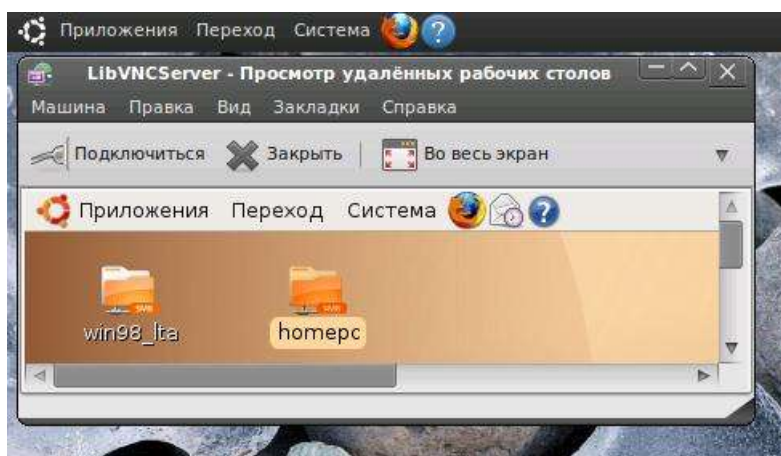


Рис. 9. Подключения с vmUbuntu к vmUbuntu10 с помощью клиента удаленных рабочих столов

Дело в том, что серверы удаленных рабочих столов Ubuntu работают по протоколу VNC, а доступ из Windows реализован на основе протокола RDP. Возникает вопрос совместимости удаленных рабочих столов Windows- и Ubuntu-машин. Один из вариантов технологии объединения удаленных рабочих столов Windows- и Ubuntu-машин состоит в том, чтобы:

- Для подключения с Ubuntu на Windows использовать rdesktop или клиента терминального сервера с протоколом RDP.
- Для подключения из Windows к Ubuntu использовать стандартные средства Windows по подключению к удаленному рабочему столу, установив для этого на Ubuntu-машине сервер RDP.

Установка RDP-сервера на Ubuntu-машину

Организовать RDP-сервер на Ubuntu можно, например, с помощью пакета **xrdp**, который поддерживает протокол удаленного рабочего стола RDP для среды XWindows, то есть для класса Linux-машин. Именно его и следует установить на Ubuntu-машине.

Рассмотрим разные варианты установок пакета **xrdp** на Ubuntu-машину. Если компьютер с Ubuntu 10.04 имеет доступ в Интернет, то достаточно войти в терминал и набрать команду:

```
sudo aptitude install gnome-rdp
```

После установки **xrdp** запустится автоматически. Если в вашей системе этого не произошло, запустите его вручную при помощи *init*-скрипта:

```
sudo /etc/init.d/xrdp start
```

Надо отметить, что этот подход доступен и в удаленном терминальном режиме, если использовать протокол SSH и утилиту PuTTY. Привлекателен этот подход тем, что можно последовательно подключаться к удаленным узлам, куда есть SSH-доступ, и там устанавливать нужное ПО.

Установив сервер RDP на Ubuntu-машину, теперь к ее рабочему столу можно подключиться из Windows, используя штатный пакет *mstsc.exe* из поставки Microsoft Windows XP SP3 (рис. 10).

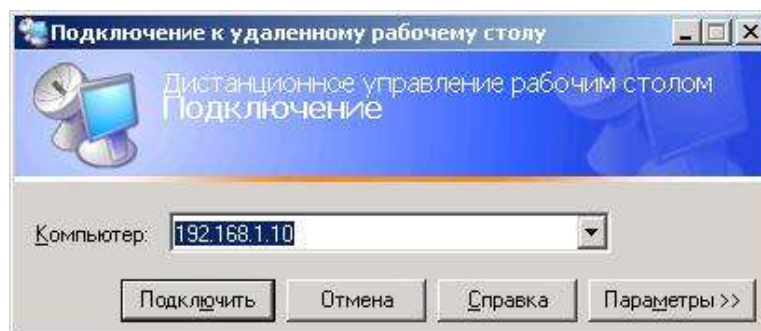


Рис. 10. Подключение из основного компьютера к рабочему столу *vmUbuntu10*

В окне «Удаленный рабочий стол» компьютера с ОС Windows появится окно RDP-сервера Ubuntu с запросом на доступ к ее удаленному рабочему столу. Для входа надо ввести имя пользователя, его пароль и нажать кнопку ОК (рис. 11).

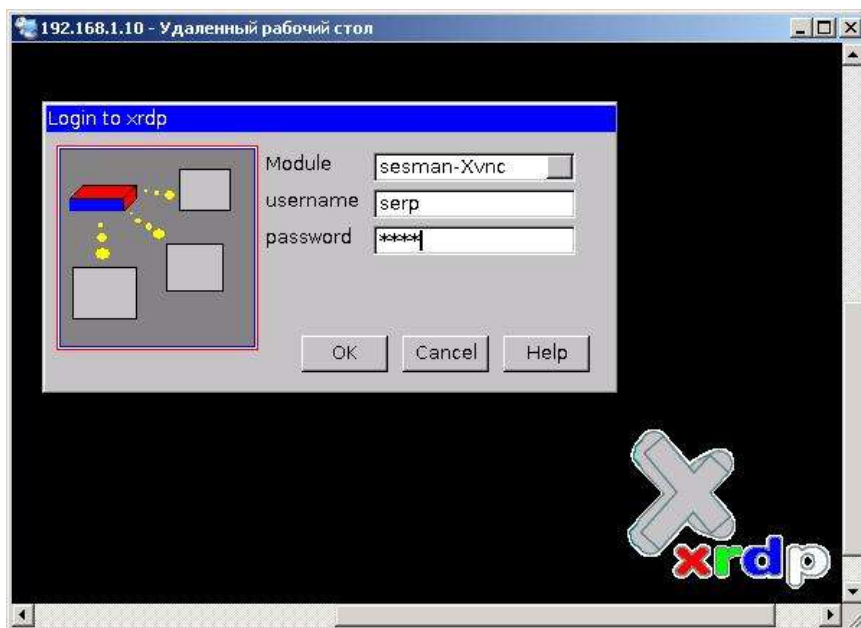


Рис. 11. Аутентификация доступа к рабочему столу vmUbuntu10

Если аутентификации прошла успешно, то на компьютере с Windows откроется окно удаленный рабочий стол Ubuntu-машины.

Однако более поздние версии, например, Ubuntu14.04 обладают такими особенностями Desktop Interface, что функциональности XRDP при доступе из Windows не хватает для проброса графики рабочего стола Ubuntu 14.04.

Возникает задача настройки альтернативного рабочего стола Ubuntu 14.04 для возможности доступа к нему стандартными средствами Windows. Для этого кроме xrdp надо установить альтернативную графическую оболочку, например, Xfce. Она будет использоваться во время удаленного подключения к компьютеру с ОС Ubuntu 14.04.

Установка и настройка xrdp и Xfce на Ubuntu 14.04

Если компьютер с Ubuntu 14.04 имеет доступ в Интернет, то достаточно войти в терминал и набрать команду:

```
sudo apt-get install xrdp
```

В ответ на запрос надо ввести пароль, после этого появится сообщение о размере пакета и для продолжения установки требуется нажать клавишу Y. По окончании процесса установки xrdp, для установки альтернативной графической оболочки надо в терминале ввести следующие команды:

```
sudo apt-get update  
sudo apt-get install xfce4
```

Далее следует указать, что для удаленной сессии будет использоваться графическое окружение xfce4. С этой целью надо в терминале ввести следующую команду:

```
echo xfce4-session > ~/.xsession
```

Для окончания процесса настройки требуется перезапустить службу xrdp путем выдачи следующей команды;

```
sudo service xrdp restart
```

Теперь можно проверить подключение компьютера с ОС Windows 7 к удаленному рабочему столу компьютера с ОС Ubuntu 14.04 (192.168.200.99), используя стандартные средства Windows (рис. 12).

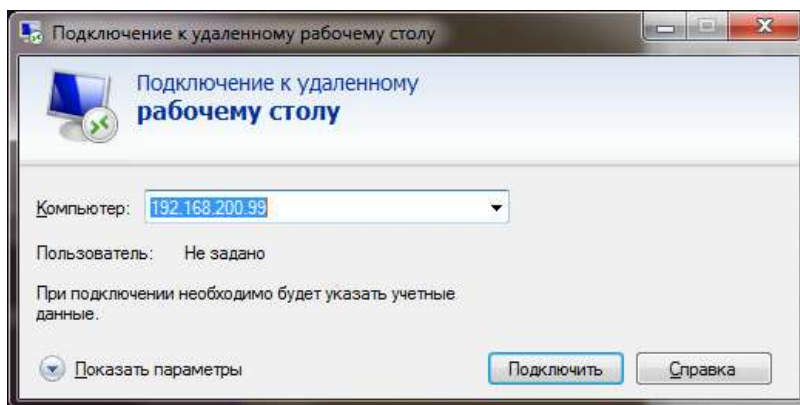


Рис. 12. Подключение к удаленному рабочему столу из Windows 7

После нажатия кнопки подключения появится окно аутентификации, аналогичное тому, что было и в более ранних версиях ОС (рис. 11), в котором надо ввести логин и пароль пользователя удаленного рабочего стола Ubuntu-компьютера. Затем появится диалоговое окно, где отображается процесса регистрации и подключения по RDP к удаленному рабочему столу (рис. 13)



Рис. 13. Протокол процесса регистрации и подключения по RDP

Если все было настроено правильно, то появится рабочий стол Ubuntu, но в графической оболочке Xfce, которая, сохраняет всю функциональность Ubuntu 14.04, доступную для конкретного пользователя (рис. 14).

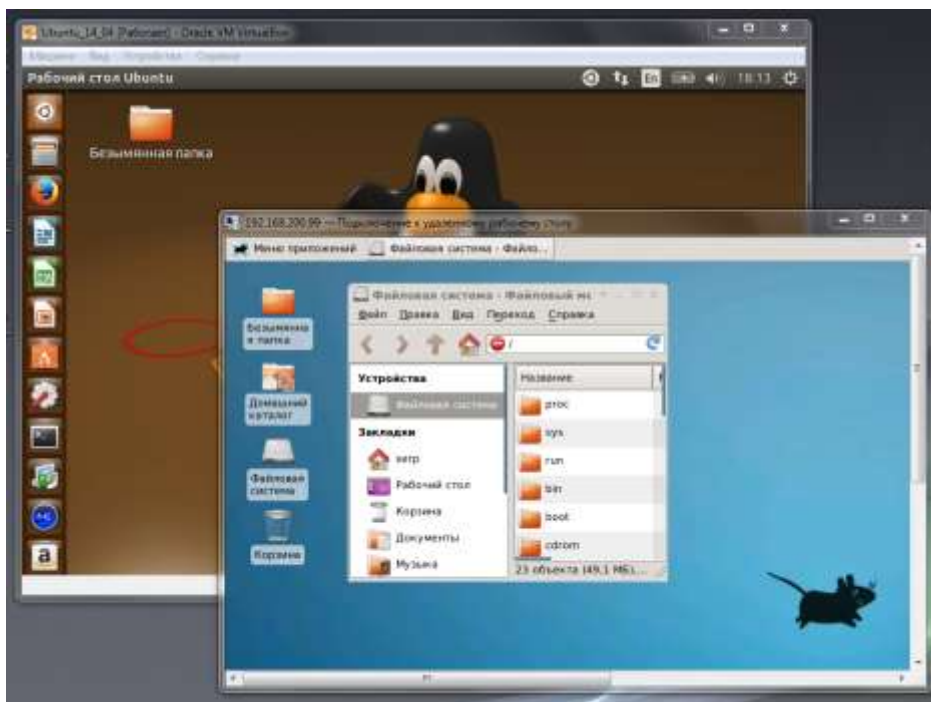


Рис. 14. Удаленный рабочий стол Ubuntu 14.04 с оболочкой Xfce

Основные выводы

1. Показана возможность организации работы с удаленными рабочими столами в Ubuntu—Windows и Ubuntu—Ubuntu системах.
2. Использование протокола RDP позволяет в Ubuntu—Windows системах удаленно конфигурировать и настраивать множество узлов сети
3. Отмечено, что удаленный компьютер с ОС Windows блокируется при входе на его удаленный рабочий стол, а в ОС Ubuntu можно работать одновременно на основном и на удаленном рабочем столе.
4. ОС Ubuntu Server 14.04 позволяет организовать одновременный доступ нескольких пользователей на один сервер, используя для каждого из них свой удаленный рабочий стол.
5. Это позволяет в компьютерном классе на ОС Windows организовать работу в среде ОС Ubuntu, при условии, что именно она и установлена на этом сервере.
6. Конфигурационный файл создания группы пользователей для работы по RDP с предоставлением каждому из них индивидуальных прав по доступу к домашним каталогам имеет вид:

```
#!/usr/bin/env bash
# Запуск этого файла --> sudo bash create-rdp-users
GroupName='remote-ws'
UserName='ws'
UserCount=3
```

```

# Создание группы пользователей для удаленной работы по xRDP
groupadd $GroupName
### cat /etc/group

# Создание папки для домашних каталогов rdp-пользователей
mkdir /home/rdp-ws

# Создание в этой группе UserCount пользователей
(ws1,...,wsN>)
for (( n=1; n <= $UserCount; n++ ))
do
    # Принудительное удаление (-f) пользователя и
    # его каталога (-r)
    userdel -f -r $UserName$n

    # Создание в rdp-группе нового пользователя и
    # его домашнего каталога
    HomeDir='/home/rdp-ws/'$UserName$n
    useradd -m -d $HomeDir -g $GroupName -G $GroupName
        $UserName$n

    # Задание пароля пользователя, аналогичного его имени
    echo "$UserName$n:$UserName$n" | chpasswd

    # Делаем пользователя "скрытым", чтобы его не было во
    # входном списке (его ID меньше 1000, начиная с 950)
    newID=$((950 + $n))
    usermod -u $newID $UserName$n

    # Профиль подключения по rdp на основе xfce4 (см. п.4)
    # echo xfce4-session > $HomeDir/.xsession
Done

# Перезапуск сервиса xrdp
sudo service xrdp restart
exit 0

```

Библиографический список

1. Хабаров С. П., Жук Ю. А. Сетевые технологии взаимодействия Ubuntu и Windows платформ // С. П. Хабаров. — СПб.: Наука и техника, 2013.— 369 с.
2. Панфилов И.В., Хабаров С. П., Заяц А.М. Информационные сети: Учебное пособие. — СПб.: СПбГЛТА, 2003.— 164 с.

С.П. Хабаров, кандидат технических наук, доцент
В.М. Амбросовский, кандидат технических наук, доцент
А.С. Корнев, ведущий специалист

ОБОСНОВАНИЕ ВЫБОРА ИНТЕРФЕЙСОВ В СИСТЕМЕ КООРДИНИРОВАННОГО УПРАВЛЕНИЯ С ИСПОЛЬЗОВАНИЕМ ГРУППОВОЙ ЭКСПЕРТНОЙ ОЦЕНКИ

Решена задача обоснования выбора интерфейсов в системе координированного управления с использованием групповой экспертной оценки объектов при непосредственном оценивании

Введение

В рамках базового проекта систем координированного управления [1, 2], структурно-функциональная схема системы закладывается на этапе эскизного проектирования и определяется технико-экономической целесообразностью. При проектировании системы координированного управления, которая впоследствии позволяет с минимальными доработками программно-аппаратного обеспечения осуществлять разработку новой системы, необходима разработка базовой структурно-функциональной схемы. Под базовой функционально-структурной схемой системы понимается такая схема, которая включает в свой состав минимальный набор блоков и может быть взята за основу для разработки любой другой функционально-структурной схемы системы координированного управления и дополнена в соответствии с требованиями, предъявляемыми к данной конкретной системе. При разработке базовой структурно-функциональной схемы требуется большой опыт и не одного человека, здесь как нельзя лучше можно воспользоваться методами групповых оценок полученных данных от экспертов.

В данной статье рассматривается обоснование интерфейсов передачи данных в структурно-функциональной схеме системы координированного управления с помощью групповой экспертной оценки объектов при непосредственном оценивании.

Постановка задачи

Базовая структурная схема должна иметь иерархическую топологию, на верхнем уровне иерархии располагаются панели ПКУ совместно с операторскими станциями, которые задают значение сигналов управления и режимов работы самой системы. Далее расположено устройство вычислительное, которое в соответствии с заложенными в него алгоритмами вырабатывает сигналы управления для средств активного управления. Связь

устройства вычислительного с системами ДАУ осуществляется через устройства сопряжения, которые располагаются на нижнем уровне иерархии. Базовая функциональная схема представлена на рис. 1.



Рис. 1. Базовая функциональная схема системы

Необходимо для данной функциональной схемы системы координированного управления обосновать выбор интерфейсов передачи данных между составными частями системы. Используемые интерфейсы для системы координированного управления приведены в табл. 1.

Т а б л и ц а 1

Использование интерфейсов в системе

Связь между	CAN	RS422/485	Ethernet	AI/AO/DI/DO
Устройством вычислительным и ПКУ	+	+	+	+
Устройством вычислительным и операторской станцией	+	+	+	-
Устройством вычислительным и устройством сопряжения	+	+	+	+
Устройством вычислительным и устройством вычислительным	+	+	+	-

Обоснование выбора интерфейсов передачи данных в системе координированного управления

Проведем обоснование интерфейсов передачи данных в системе координированного управления с помощью метода групповых оценок объек-

тов при непосредственном оценивании полученных данных от экспертов. Для этого необходимо проанализировать опыт экспертов из этой области, а далее воспользоваться методами групповых оценок полученных данных от экспертов [3].

Использование методов групповых оценок позволяют получить достоверные результаты в случае хорошо подобранной группы экспертов и согласованности их мнений [3]. Поэтому для оценки требований к использованию интерфейсов в системах координированного управления был проведен опрос экспертов в компаниях, занимающихся разработкой и производством интегрированных мостиковых систем и систем координированного управления. Результаты опроса 9 экспертов приведены в табл. 2,3,4,5. Как следует из теории обработки наблюдений, такого количество экспертов достаточно, чтобы сделать выводы с достоверностью не ниже 70%, дальнейшее увеличение количества экспертов не вносит новых данных, но может привести к ухудшению статистики [3].

В приводимых ниже таблицах проведена оценка использования интерфейсов в системах координированного управления девятью экспертами.

Представим результаты оценивания, приведенные в таблицах, величинами $x_{i,j}^h$, где i - номер объекта, j - номер эксперта, h - номер показателя. Например, данные из табл. 3 ($h = 2$) для интерфейса RS422/485 ($i = 3$), сделанные пятым экспертом ($j = 5$), могут быть записаны следующим образом $x_{3,5}^2 = 0,2$.

Т а б л и ц а 2

Интерфейс между устройством вычислительным и панелями ПКУ

<i>Интерфейс</i>	<i>Эк-т 1</i>	<i>Эк-т 2</i>	<i>Эк-т 3</i>	<i>Эк-т 4</i>	<i>Эк-т 5</i>	<i>Эк-т 6</i>	<i>Эк-т 7</i>	<i>Эк-т 8</i>	<i>Эк-т 9</i>
CAN	0,5	0,7	1	0,5	0,8	0,5	0,7	0,3	0,5
RS422/485	0,5	0,2	0	0,5	0,2	0,5	0,3	0,7	0,5
Ethernet	0	0	0	0	0	0	0	0	0
AI/AO/DI/DO	0	0,1	0	0	0	0	0	0	0

Т а б л и ц а 3

Интерфейс между устройством вычислительным и операторской станцией

Интерфейс	Эк-т 1	Эк-т 2	Эк-т 3	Эк-т 4	Эк-т 5	Эк-т 6	Эк-т 7	Эк-т 8	Эк-т 9
CAN	0	0	0	0	0	0	0	0	0
RS422/485	0,5	0,3	0,5	0,5	0,2	0,5	0,3	0,6	0,5
Ethernet	0,5	0,7	0,5	0,5	0,8	0,5	0,7	0,4	0,5

Т а б л и ц а 4

Интерфейс между устройством вычислительным
и устройством сопряжения

Интерфейс	Эк-т 1	Эк-т 2	Эк-т 3	Эк-т 4	Эк-т 5	Эк-т 6	Эк-т 7	Эк-т 8	Эк-т 9
CAN	0,5	1	0	0,5	0,8	0,5	0,7	0	0,5
RS422/485	0,5	0	0	0,5	0,2	0	0,3	0,6	0,5
Ethernet	0	0	1	0	0	0,5	0	0,4	0
AI/AO/DI/DO	0	0	0	0	0	0	0	0	0

Таблица 5

Интерфейс между устройствами вычислительными

Интерфейс	Эк-т 1	Эк-т 2	Эк-т 3	Эк-т 4	Эк-т 5	Эк-т 6	Эк-т 7	Эк-т 8	Эк-т 9
CAN	0	0	0	0	0	0	0	0	0,1
RS422/485	0,1	0	0	0,2	0	0,3	0,3	0,2	0
Ethernet	0,9	1	1	0,8	1	0,7	0,7	0,8	0,9

В качестве групповой оценки для каждого из интерфейсов можно принять среднее взвешенное значение его оценки [3]:

$$x_i = \sum_{k=1}^l \sum_{j=1}^m q_k x_{i,j}^k k_j, (i = 1, 2 \dots n) \quad (1)$$

где, q_k – коэффициенты весов показателей сравнения объектов;
 k_j – коэффициенты компетентности экспертов.

Проведем в качестве примера расчет для табл. 3. В качестве начальных условий ($t = 0$, начальный шаг расчета) уровни компетентности всех экспертов принимаются одинаковыми:

$$k_j^0 = \frac{1}{m} \quad (j = \overline{1, m}) \quad (2)$$

где, m – количество экспертов.

Далее для расчета используются следующие рекуррентные соотношения [3]:

$$x_i^t = \sum_{j=1}^m x_{ij} k_j^{t-1}, \quad (i = \overline{1, n}) \quad (3)$$

$$\lambda^t = \sum_{i=1}^n \sum_{j=1}^m x_{ij} x_i^t \quad (4)$$

$$k_j^t = \frac{1}{\lambda^t} \sum_{i=1}^n x_{ij} x_i^t, \quad (j = \overline{1, m-1}) \quad (5)$$

$$k_m^t = 1 - \sum_{j=1}^{m-1} k_j^t \quad (6)$$

$$\max | \mathbf{[(x)]}_{i^t} - x_{i^t}(t-1) | < E \quad (7)$$

где, x_i^t – групповая оценка на для i -го объекта на t -м шаге на основе индивидуальных оценок x_{ij} ;

λ^t – нормировочный коэффициент;

k_j^t – коэффициент компетентности i -го эксперта на t -м шаге;

k_m^t – коэффициент компетентности i -го эксперта из условия нормировки.

Окончание расчета определяется признаком окончания итерационного процесса, который определяется выражением (7).

Для варианта, соответствующего табл. 3, начальный уровень компетентности всех экспертов принимаем в соответствии с (2)

$$k_j^0 = \frac{1}{9} \quad (j = \overline{1, 9}).$$

Рассчитываем групповую оценку для 1-го шага в соответствии с (3):

$$x_1^1 = 0;$$

$$x_2^1 = \left(\frac{0,5 + 0,3 + 0,5 + 0,5 + 0,2 + 0,5 + 0,3 + 0,6 + 0,5}{9} \right) = 0,433;$$

$$x_3^1 = \left(\frac{0,5 + 0,7 + 0,5 + 0,5 + 0,8 + 0,5 + 0,7 + 0,4 + 0,5}{9} \right) = 0,567.$$

Рассчитываем нормировочный коэффициент для 1-го шага в соответствии с (4):

$$\lambda^1 = 0 + 0,433 \cdot (0,5 + 0,3 + 0,5 + 0,5 + 0,2 + 0,5 + 0,3 + 0,6 + 0,5) + 0,567 \cdot (0,5 + 0,7 + 0,5 + 0,5 + 0,8 + 0,5 + 0,7 + 0,4 + 0,5) = 4,58;$$

Рассчитываем компетентности экспертов для 1-го шага в соответствии с (5) и (6):

$$k_1^1 = k_2^1 = k_3^1 = k_4^1 = k_5^1 = \frac{1}{4,58} (0 + 0,5 \cdot 0,433 + 0,5 \cdot 0,567) = 0,109;$$

$$k_{\frac{1}{2}} = k_{\frac{1}{7}} = 0,115;$$

$$k_{\frac{1}{5}} = 0,118;$$

$$k_{\frac{1}{8}} = 0,107;$$

$$k_{\frac{1}{9}} = 1 - (4 \cdot 0,109 + 2 \cdot 0,115 + 0,118 + 0,107) = 0,109.$$

Далее задаем необходимую нам точность E для окончания итерационного процесса, повторяем расчет для 2-го шага и проверяем условие (7), если оно не выполняется, то повторяем расчет для 3-го шага и так далее.

Окончательные результаты расчета групповой экспертной оценки при непосредственном оценивании требований к использованию интерфейсов в системах координированного управления с использованием данных, представленных выше, приведены в табл. 6.

Таблица 6

Сравнительный анализ использование интерфейсов

Связь между	CAN	RS422/485	Ethernet	AI/AO/DI/DO
Устройством вычислительным и ПКУ	0,74	0,26	0	0
Устройством вычислительным и операторской станцией	0	0,43	0,57	0
Устройством вычислительным и устройством сопряжения	0,57	0,28	0,15	0
Устройством вычислительным и устройством вычислительным	0	0,09	0,91	0

На основе данных, полученных в табл. 6, в базовой функционально-структурной схеме для связи устройства вычислительного с устройством сопряжения, а также с ПКУ нужно использовать CAN интерфейс, для связи устройства вычислительного с операторской станцией нужно ориентироваться на Ethernet, но при этом необходимо предусмотреть техническую возможность использования RS422/485. При организации горячего резервирования необходимо использовать Ethernet.

Заключение

Использование методов групповой экспертной оценки объектов позволяют получить достоверные результаты, поэтому их можно использовать для обоснования выбора в случае, когда сложно сформулировать четкие критерии выбора. В данной статье был приведен пример обоснования выбора интерфейсов в системе координированного управления. В подтверждение полученных результатов можно привести две различные системы координированного управления «NautoSteer AS» ведущей немецкой фирмы «Raytheon» и «NavDP4000» ведущей финской компании «Navis Engineering». Данные системы для связи своих составных частей используют те

же интерфейсы, что мы обосновали для применения в системе координированного управления.

Библиографический список

1. Амбросовский В.М., Баглюк Ю.В., Коданев А.Н., Коренев А.С. Система координированного управления движением корабля. Морской вестник — СПб.: МорВест, №2, 2011. - с.85-87.
2. Хабаров С.П., Коренев А.С. Информационная структура алгоритмического обеспечения системы координированного управления движением судна. //Сборник научных трудов. Информационные системы и технологии: Теория и практика — СПб.: СПбГЛТУ, 2011. №3 ч.2 – с.30-37.
3. Хабаров С.П. Интеллектуальные информационные системы. http://www.habarov.spb.ru/new_es/index.htm

Н.П. Васильев кандидат технических наук, доцент

КОНЕЧНЫЕ АВТОМАТЫ ДЛЯ ЗАДАЧ РАСШИФРОВКИ И ПРОВЕРКИ НАУЧНЫХ НАЗВАНИЙ РАСТЕНИЙ

Введение. Точные научные названия растений могут оказаться достаточно ёмкими, например: *Aster ageratoides Turcz. var. harae (Makino) Kitam. f. yezoensis (Kitam. et Hara) Ohwi cv. Hortensis*. Проверка правильности написания таких названий, с выделением в этих названиях составных частей – соподчиненных таксономических единиц, имен их авторов, является актуальной задачей, особенно когда требуется автоматическая компьютерная обработка. Например, результаты такого разбора названий растений могут быть входными данными для поиска в базах данных, содержащих информацию о таксонах.

Для решения этой задачи в статье предлагается использовать аппарат конечных автоматов. Прежде чем изложить суть предлагаемого подхода следует кратко описать правила, регулирующие международные научные названия растений, которые изложены в «Международном кодексе ботанической номенклатуры».

Названия растений, как правило, образуются из слов латинского языка (впрочем, допускаются слова любого другого языка, преимущественно древнегреческого). В 1753 году вышла книга Карла Линнея «*Species Plantarum*». В ней он положил начало бинарным названиям растений, состоящим из названия рода и видового эпитета.

Вид является основным таксоном как дикорастущих, так и культурных растений; в его пределах могут различаться таксоны более низкого ранга: подвид (*subspecies*, сокращенно *subsp.*), разновидность (*varietas* –

var.), форма (forma – f.). Основным внутривидовым таксоном культурных растений, отличающихся от исходного дикорастущего предка признаками, приобретенными в культуре, служит сорт (культивар); сходные сорта могут быть объединены в группы сортов (сортогруппы). Внутривидовые таксоны, кроме сортов, подобны видовым эпитетам и пишутся обычно в форме прилагательного после сокращенного обозначения таксона (subsp., var., f.). Названия внутривидовых таксонов пишутся после видового названия в порядке возрастающей дробности: подвид, разновидность, форма, культивар.

Название сорта после 1 января 1959 г. может быть дано на любом языке, кроме латинского; присвоенные до этой даты латинские названия сортов сохраняются. Название сорта пишется всегда с заглавной буквы, включая все слова этого названия, и заключается в одинарные кавычки или перед ним ставится сокращение слова «культивар» – cv.

После латинского (или латинизированного) названия таксона может быть приведена фамилия его автора, обычно сокращенно: так, авторство К. Линнея обозначается буквой L., В. Л. Комарова – Ком. Один из важнейших принципов номенклатуры – принцип приоритета: обязательно название самое раннее начиная с 1753г. По этой причине, если - несколько авторов, то автор старого названия указывается в скобках перед автором нового. Например, *Convallaria multiflora* L. переименована в *Polygonatum multiflorum* (L.) All.

Конечный автомат как способ описания языка научных названий растений. Выше были изложены простейшие правила формирования научных названий растений. Эти правила можно рассматривать с абстрактных позиций определения формального языка [1]. В этом определении, как известно, фигурирует некоторое множество символов, называемое алфавитом, из которых можно формировать строки – слова. Если эти строки формируются в соответствии с некоторыми правилами, то в результате, получится формальный язык, содержащий некоторое под множество таких строк.

Конечные детерминированные автоматы представляют собой очень удобный и простой способ описания формальных языков, который фактически определяет алгоритм распознавания слов этого языка. К сожалению, не все языки можно описать с помощью этого аппарата. К счастью, язык научных названий растений к этой категории не относится и его можно представить с помощью классического автомата. Приведем формальное описание такого автомата, решающего задачу распознавания научных названий растений.

Как известно, детерминированным конечным автоматом называется объект, в котором можно выделить следующие пять элементов:

- Конечное множество состояний $Q = \{q_1, q_2, \dots, q_n\}$.

- Конечное множество символов входной ленты (алфавит)
 $\Sigma = \{a_1, a_2, \dots, a_m\}$.

- Начальное состояние q_s – некоторый элемент множества Q , выделяемый особо.

- Множество допускающих состояний $F = \{f_1, f_2, \dots, f_k\}$. Это также обычные состояния множества Q , выделенные в самостоятельный набор.

- Наконец, функция переходов $\delta(q_1, a) = q_2$, сопоставляющая любой паре: состояние, символ некоторое новое состояние. Подобные переходы автомата из одного состояния в другое, обычно записывают так: состояние, символ->состояние.

В этой абстрактной схеме для решения поставленной задачи под множеством символов будем понимать:

- названия таксонов и фамилии авторов – произвольные строки из латинских (или иных, например, русских букв) и ряда специальных знаков, которые могут встречаться в фамилиях авторов (например, точка); такие строки в алфавите языка будут представлять один и тот же символ, который условно обозначим – str.

В алфавит также входит ряд вспомогательных слов, употребляемых в спецификациях растений, а именно:

- ssp. или subsp. – ключевое слово, употребляемое перед названием подвида (subspecies);

- var. – ключевое слово, употребляемое перед названием разновидности (varieties);

- f. – ключевое слово, употребляемое перед названием формы (forma);

- скобки (и) – используются для указания прежнего автора таксона;

- cv. – ключевое слово, употребляемое перед названием культивара (cultivar);

- кавычки одинарные - ‘или двойные -“ – используются для указания сорта.

Распознавание названных выше символов возлагается на специальный программный модуль, который традиционно называется лексическим анализатором. В данном случае это довольно простой модуль, который разбивает исходное название растения на отдельные слова. Словом также являются скобки и кавычки. Разделителем является один или несколько пробелов. Чтобы выделить скобки, апострофы и кавычки, последние предварительно окаймляются пробелами. Затем, очередная часть названия (получившиеся в результате разбиения) определяется как один из перечисленных символов и по запросу поставляется на входную ленту автомата. Проверка выделенных слов начинается с сопоставления с ключевыми словами, иначе последние будут восприняты как обычные строки – str. Если

все выделенные слова исчерпаны, то на входную ленту автомата выдается символ – end.

Состояния автомата и возможные переходы.

Автомат содержит достаточно большое количество состояний, а именно 44 состояния. К этим состояниям следует добавить еще два допускающих состояния: success - название распознано успешно, failure - название не распознано. В формате данной статьи не имеет смысла описывать все состояния, поэтому далее приводится лишь их часть. В случае если, для какого-то состояния с входной ленты читается символ, который не предусмотрен в списке переходов для данного состояния, то считается, что автомат переходит в ошибочное финишное (допускающее) состояние - failure. То есть спецификация не распознана.

0: - Это начальное состояние, когда не распознана ни одна таксономическая единица. Состояние соответствует ожиданию прихода с входной ленты, от лексического анализатора, названия рода, которая лексическим анализатором трактуется как обычная строка-str. Возможен только один переход из этого состояния: 0,str->1.

1: Распознан род, ожидается названия вида. Возможен один переход из этого состояния: 1,str->2.

2: Распознан вид. После вида может быть указан автор, либо прежний автор в скобках. Если автор (авторы) не приводятся, то далее могут быть указаны соподчиненные таксономические единицы, а именно, либо подвид – subsp, либо разновидность – var, либо форма – f, либо культивар – cv, наконец, может быть указан и сорт, то есть кавычки – одинарные или двойные. Согласно сказанному возможны следующие переходы:



Рис. 1. Схема работы распознавателя названий растений.

2,str->21
 2,(->3
 2,susp->5
 2,var->9
 2,f->13
 2,cv->17
 2,'->211
 2,end->success.

21: Распознано имя автора вида, или его часть. В последнем случае, с входной ленты возможно получение очередной «порции» имени автора. Иначе, в этом состоянии ожидается указание соподчиненной таксономической единицы. В результате, возможны следующие переходы:
 21,str->21
 2,susp->5
 2,var->9

2,f->13

2,'->211.

3: Распознана открывающаяся скобка, после которой должен быть указан прежний автор вида. В этом состоянии ожидаются строки, содержащие имя автора. Таким образом, возможен только один переход: 3,str->31.

31: Распознано имя прежнего автора вида, или его часть. Этот автор указывается в скобках.

В этом состоянии возможно продолжение имени, или закрывающаяся

скобка:

31,str->31

31,)->32

32: Распознан прежний автор вида, который указан в скобках. В этом случае должен быть обязательно указан современный автор вида, без скобок. В результате, возможен только один переход: 32,str->4.

4: Распознан род и вид и авторы вида, если были указаны. В этом состоянии ожидается название соподчиненной таксономической единицы:

4,susp->5

4,var->9

4,f->13

4,cv->17

4,'->211

2,end->success.

Прочие состояния и связанное с ними поведение автомата в основном похоже на его поведение, приведенное выше. Например, в пятом состоянии будет ожидаться строка с названием подвида, после которого могут быть указаны авторы, либо соподчиненная таксономическая единица (разновидность, форма и т. д.). Аналогично, в девятом состоянии - будет ожидаться строка с указанием разновидности, за которой могут быть указаны авторы, либо следующая по дробности таксономическая единица (форма, культивар, сорт).

Заключение. Описанный подход положен в основу окончательного алгоритма и программы для проверки и поиска заданного названия (названий) растения (растений) в базе данных. База данных в данной статье не обсуждается. Практическое использование программы подтверждает правильность и эффективность предложенного подхода. На рисунке 2 представлен фрагмент интерфейса программы, где для заданного названия растения представлены выделенные в этом названии таксономические единицы, а также результат поиска этого растения в дереве таксонов.

О проекте | Дерево таксонов | Поиск

Berberis thunbergii cv. Atropurpurea

	Название	Автор
Отдел	MAGNOLIOPHYTA	
Класс	MAGNOLIOPSIDA	
Подкласс		
Порядок	BERBERIDALES	
Семейство	BERBERIDACEAE	
Род	Berberis	L.
Вид	thunbergii	DC.
Подвид		
Разновидность		
Форма		
Культивар	Atropurpurea	
Сорт		

Рис. 2. Фрагмент интерфейса программы

Библиографический список

1. Мозговой М. В. Классика программирования: алгоритмы, языки, автоматы, компиляторы. Практический подход. – СПб.: Наука и техника, 2006. – 320с.

Ю.А. Жук кандидат педагогических наук, доцент;
Д.А.Плотников магистр, ИТМО

КЛАССИФИКАЦИЯ УРОВНЯ ЗНАНИЙ СТУДЕНТОВ НА ОСНОВЕ МЕТОДОВ МАШИННОГО ОБУЧЕНИЯ

Система высшего образования располагает огромными объемами данных о студентах. Эти данные могут храниться в общей университетской базе данных, если ВУЗ обладает центром дистанционного обучения, либо другой структурой централизованной контрольной системы знаний студентов. Даже в случае неразвитой цифровой инфраструктуры всегда есть возможность

провести среди студентов анкетирование по определенным вопросам, для получения необходимой информации.

Одно из популярных направлений мониторинга и диагностики различных образовательных показателей – это использования методов статистического анализа для принятия решений и прогнозирования. С помощью этих методов можно выделить из больших объемов данных о студентах, которые накоплены в автоматизированных системах вуза или получены другими способами, ранее неизвестные знания или закономерности, которые могли помочь в повышении эффективности образовательного процесса. Эти данные могут быть различными: скрытые паттерны, зависимости между параметрами, которые до этого момента не были замечены, прогнозы.

Ряд исследований посвящён теме использования методов машинного обучения для прогнозирования успехов студентов в обучении. В большинстве из них исследователи пытаются найти эффективный алгоритм для прогнозирования.

Целью, практически всех научных работ на данную тему, является выбор наиболее эффективного метода или алгоритма диагностики и прогнозирования. Входными данными являются множества параметров, которые характеризуют обучающегося. Чаще всего для получения таких данных используется метод анкетирования. Анкета или опросник включает в себя, помимо множества характеристик, значение некоторого параметра, который исследователи намерены прогнозировать. Общее количество анкет делится на десять частей – девять из них используются в качестве обучающей выборки, а десятая служит для тестирования. Каждый алгоритм обучается и тестируется, показателем, на основе которого происходит сравнение эффективности алгоритмов, является точность прогнозирования или классификации.

В работе [1] португальские исследователи из University of Minho поставили целью исследования ответить на вопросы «Возможно ли предсказать, чего достигнут студенты в обучении?» и «Какие факторы на эти достижения могут повлиять?», как итог прогнозировались результаты итогового теста по предмету. Для каждой модели были подготовлены три входных набора (включающие и не включающие школьные оценки). Так же были протестированы несколько алгоритмов машинного обучения: Наивный Байес (NB), Нейронная Сеть (NN), Метод Опорных Векторов (SVM) и Случайный Лес (RF). Для реализации алгоритмов исследователями был выбран язык R, с использованием открытой библиотеки RMiner. Лучшие результаты показал метод RF(случайных деревьев), следующим за ним идет Naive Predictor (NB). Алгоритмы SVM и Neural Networks, показали менее точные результаты. Что именно послужило причиной такого распределения мест в исследовании не анализируется, но предполагается, что количество обучающих сессий и объемы выборки могли повлиять на показатели SVM и NN в худшую сторону.

В следующем исследовании [2] были собраны данные о 257 студентах, на основе которых ими были построены три модели по прогнозированию результатов студентов в итоговом тесте. Параметры каждой записи студента состояли не только из обычных для таких исследований полей, таких как пол, возраст, оценки в школе и др., но и из тех, что встречаются не так часто – наличие интернета и объем учебных материалов в распоряжении студента. Авторы работы сравнивали три различных алгоритма машинного обучения: наивный Байес, нейронные сети и одну из модификаций дерева решений. Все алгоритмы показали примерно равную точность – 71,2% у нейронной сети, 73,93% у дерева решений и 76,65% у наивного Байеса.

Самое масштабное по количеству обработанных анкет студентов (более 21 тысячи) исследование [3] было проведено в университете Хильдесхайм (Германия). Задачей исследования было протестировать алгоритмы машинного обучения на точность прогнозирования результатов обучения. Среди этих алгоритмов были: Наивный Байес (NB), Метод Опорных Векторов (SVM) и Дерево Решений (DT). Результаты, которые были получены в этом исследовании, показали: наивный Байес дал менее эффективные показатели (максимальная точность 0.57) по сравнению с SVM (0.77) или DT (0.62). По мнению авторов исследования, стоит учесть вариативность подхода к оцениванию эффективности алгоритмов.

Таким образом, анализ некоторых исследований из области применения алгоритмов машинного обучения в прогнозировании успехов студентов показывает, что эта тема достаточно актуальна на данный момент. Во-первых, сама задача имеет потенциал для применения в системе ВО. Во-вторых, алгоритма решающего данную задачу с точностью большей чем другие не найдено, что представляет перспективу для дальнейших исследований.

Идея создания системы прогнозирования результатов обучения, на основе методов машинного обучения, представляется актуальной задачей. Для разработки такой прогнозирующей системы необходимо собрать данные о студентах, аргументировано провести выбор алгоритма классификации, обучить на данных алгоритм, затем его протестировать и если алгоритм показывает приемлемые показатели точности классификации, то использовать его для прогнозирования.

Естественно возникает вопрос: Какие данные следует использовать? Конкретного ответа в виде списка параметров нет. Могут понадобиться любые данные, так или иначе характеризующие студента и, вероятно, влияющие на его способности к обучению, научные успехи и т.д. Среди множества данных можно выделить пол, возраст поступления в ВУЗ, баллы за ЕГЭ, баллы за экзамены и личностные качества, наличие задолженностей в настоящем и прошлом, работает ли студент помимо учебы или же нет и др. Некоторыми исследователями отмечается, что на результаты прогнозирования в значительной степени могут влиять неочевидные данные, такие как уровень образования родителей или доход семьи студента и мно-

гие другие факторы. Таким образом, следует получить выборку определенного количества студентов, где каждый экземпляр характеризуется набором некоторых параметров, признаков, а мы должны определить класс, к которому этот студент будет отнесен. В качестве метки класса может являться диагностика: «сдаст экзамен» или «не сдаст экзамен», «поступит или нет в магистратуру», «получит или нет стипендию» и т.д.

Анализ исследований связанных с использованием машинного обучение в прогнозировании результатов студентов показал, что в среде исследователей нет определенного мнения о четком преимуществе одного или нескольких алгоритмов над остальными. Одним из направлений дальнейшего исследования данной темы можно предложить протестировать композиционные алгоритмы или бустинг для классификации уровня знаний студентов.

Наиболее известный метод, основанный на комбинировании - это бустинг. Бустинг - это метод для повышения точности классификации «слабых» классификаторов, которые объединены в один составной, «сильный» классификатор. Одним из самых популярных методов бустинга является AdaBoost (Адаптивный бустинг), предложенный [4]. AdaBoost является достаточно общим алгоритмом бустинга, который может быть использован в сочетании со многими другими алгоритмами обучения для повышения их эффективности с помощью итерационного процесса. Некоторые модификации алгоритма AdaBoost рассмотрены и предложены в работах [5, 6, 7].

Первоначально, одинаковые веса $h = (1/n, \dots, 1/n)$ назначаются всем примерам. В каждой итерации, веса всех ошибочно классифицированных примеров увеличиваются, а веса правильно классифицированных примеров снижаются (см. алгоритм 1). Как следствие, слабый классификатор вынужден сосредоточить внимание на «трудных» примерах обучающей выборки, выполняя дополнительные итерации и создавая новые классификаторы. Кроме того, вес α_i присваивается каждому классификатору. Этот вес характеризует общую точность классификатора и является функцией от суммарного веса $e(t)$ и правильно классифицированных примеров. Он рассчитывается с учетом распределения h_i . Вес α_i характеризует также значимость классификатора $c(t)$. Таким образом, большие значения весовых коэффициентов приписываются более точным классификаторам. Эти коэффициенты используются для классификации новых примеров. Распределение h_i обновляется с помощью правила, показанного в алгоритме [алгоритм 1]. Эффект этого правила заключается в повышении весов примеров, которые неправильно классифицированы и уменьшении весов правильно классифицированных примеров. Таким образом, вес имеет тенденцию концентрироваться на "плохих" примерах. Итоговый классификатор представляет собой взвешенную сумму результатов классификации T слабых классификаторов.

Основным преимуществом AdaBoost по сравнению с другими методами бустинга является его адаптивность. AdaBoost может использовать преимущества слабых классификаторов, которые являются более точными на самом деле, чем это предполагалось априори. Адаптация является очень важной особенностью AdaBoost и других комбинированных алгоритмов. Она достигается путем изменения весов всех ошибочно и правильно классифицированных примеров, в соответствии с некоторым правилом. На самом деле, равномерное распределение вероятностей (равные веса присвоенные всем примерам) заменяется другим распределением вероятностей в каждой итерации алгоритма, для того, чтобы улучшить точность классификации в следующей итерации. Веса примеров, в методе бустинга, можно рассматривать как дискретное распределение вероятностей на элементах обучающей выборки. Алгоритм поиска подходящего распределения вероятностей, начинается с равномерного распределения. Модифицированное распределение вероятностей может быть произвольным, т. е. множество возможных распределений вероятностей не ограничено. С одной стороны, этот произвол можно рассматривать как способ поиска оптимального решения, и это положительная черта алгоритма AdaBoost. С другой стороны, это является одним из недостатков этого подхода. Многими авторами было отмечено, что основной причиной неудовлетворительных результатов классификации AdaBoost при большом зашумлении является то, что алгоритм назначает слишком большие веса нескольким плохо классифицируемым примерам. Это приводит к переобучению. Кроме того, алгоритм не учитывает возможную дополнительную информацию об обучающей выборке. Например, число примеров в обучающей выборке и доверительные вероятности, что может дать полезную информацию, которая ограничивает множество распределений вероятностей.

АЛГОРИТМ 1: AdaBoost

ВХОД

обучающая выборка $T = ((x_1, y_1), \dots, (x_N, y_N))$ длины N , число шагов M

НАЗНАЧИТЬ

начальные веса $\alpha_i = \frac{1}{N}, i = 1, \dots, N$

ЦИКЛ:

Для каждого шага $j = 1, \dots, M$

Обучить «слабый» классификатор h^j на выборке T с весами α

Вычислить его взвешенную 0-1-ошибку $E = \sum_{i|h^j(x_i) \neq y_i} \alpha_i$

если $E=0$ **тогда** выход из цикла — $f(\cdot) = h(\cdot)$ безошибочный классификатор

Вычислить коэффициент $c_j = \ln \frac{1-E}{E}$;

Для каждого ошибочно классифицированного вектора $(h^j(x_i) \neq y_i)$

Домножить его вес α_i на $e^{c_j} = \frac{1-E}{E}$

Отнормировать пересчитанные веса, чтобы их сумма $\sum_{i=1}^M \alpha_i$ равнялась 1

ВЫХОД: классификатор $f(\cdot) = \text{sign} \sum_{j=1}^M c_j h'(\cdot)$

Разработка системы прогнозирования на основе алгоритмов AdaBoost имеет широкий исследовательский потенциал. Полученные результаты и технологии могут быть применены в практике работы образовательных учреждений в качестве дополнительного источника прогнозной информации об абитуриентах и студентах при принятии решений о приеме и выборе специализации, а также непосредственно в учебном процессе. Это дает возможность действительно реальной возможности научного обоснования методов воспитания и обучения, принимаемых решений, определенных воздействий, а также повышение качества усвоения студентами учебного материала, усиление ответственности преподавателей за развитие их научных способностей и стимулирование профессионального роста будущих специалистов. Управление в образовательной деятельности, как и управление в любой другой деятельности, предполагает прогноз и может быть осуществлено на его основе.

Библиографический список

1. Acharya A., Sinha D. Early Prediction of Students Performance using Machine Learning Techniques. International Journal of Computer Applications, 2014. V.107(1). Pp. 37-43.
2. Edin Osmanbegović, Mirza Suljic. Data mining approach for predicting student performance. Journal of Economics & Business/Economic Review, 2012.V. 10(1). Pp. 3-12
3. Thai-Nghe N., Busche A., Schmidt-Thieme L. Improving Academic Performance Prediction by Dealing with Class Imbalance // Intelligent Systems Design and Applications, 2009. ISDA '09. Pp. 878 – 883.
4. Freund Y., Schapire R.E. A decision theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences, 55(1):119-139, 1997.
5. Ferreira A., Figueiredo M., Boosting algorithms: a review of methods, theory, and applications, in: C. Zhang, Y. Ma (Eds.), Ensemble Machine Learning: Methods and Applications, Springer, New York, 2012, pp. 35–85.
6. Utkin L.V., Zhuk Y.A. Robust boosting classification models with local sets of probability distributions // Knowledge-Based Systems, Elsevier, V 61 (7), 2014.- pp. 59-75. 2014
7. Жук Ю.А. Модификация метода бустинга для робастной классификации на основе модели неопределенности второго порядка / Труды 17-ой

Н.В. Лушкин, кандидат технических наук, доцент

РЕАЛИЗАЦИЯ АЛГОРИТМА ОПРЕДЕЛЕНИЯ ЧИСЕЛ БЕТИ В ГРАФИЧЕСКИХ ОБЪЕКТАХ

Введение

Понятие структуры и формы в биологии является одним из главных. Структура теснейшим образом связана с функциональной деятельностью биологических систем, поэтому одно из основных научных направлений в биологии – морфология, занимается исследованием структуры. По существу, биологический организм представляет собой совокупность клеток, образующих ткани, которые выполняют водопроводящую, запасную и механическую функцию. Мера порядка, которую исследователь пытается внести в эти сложнейшие объекты, требует знания длин путей, величин поверхностей, объемов и форм.

Исследование структурной организации биологического объекта осложнено значительными трудностями, поэтому к решению подобных задач привлекается не только самая совершенная техника, но и математические методы и приемы, позволяющие расшифровать закодированные строения биологических объектов. Цель таких методов - преодолеть методические и аппаратные ограничения. Биологические объекты по своей природе трехмерны, и, как правило, исследователь имеет доступ к плоским модификациям объемной структуры. Таким образом, вначале необходимо изучить характеристики плоских сечений, которые могут представлять интерес. К ним относятся количество связных объектов, количество вложенных друг в друга объектов, их площади, их границы, длина и кривизна границ.

В статье обсуждается программная реализация получения чисел Бетти и некоторых других характеристик на плоскости графических объектов, которые получаются в результате сечения трехмерных структур плоскостью, когда в плоскости сечения сохраняются следы – отображения внутреннего строения. Различие способов описания структур состоит в использовании различных математических понятий [1]. При этом биологические объекты могут состоять из шаров, цилиндров, эллипсоидов вращения, волокон (рис. 1) или множества объектов разных форм (рис.2) [2]. Для однородных объектов по количественному анализу можно определить число объектов в единице объема, зная количество сечений и геометрические размеры сечений объектов измеренных на срезе. При решении подобных задач сталкиваются со сложными интегральными уравнениями, при этом

возникают задачи, которые решаются в обратном порядке, т.е. по следствию необходимо выяснить причину.

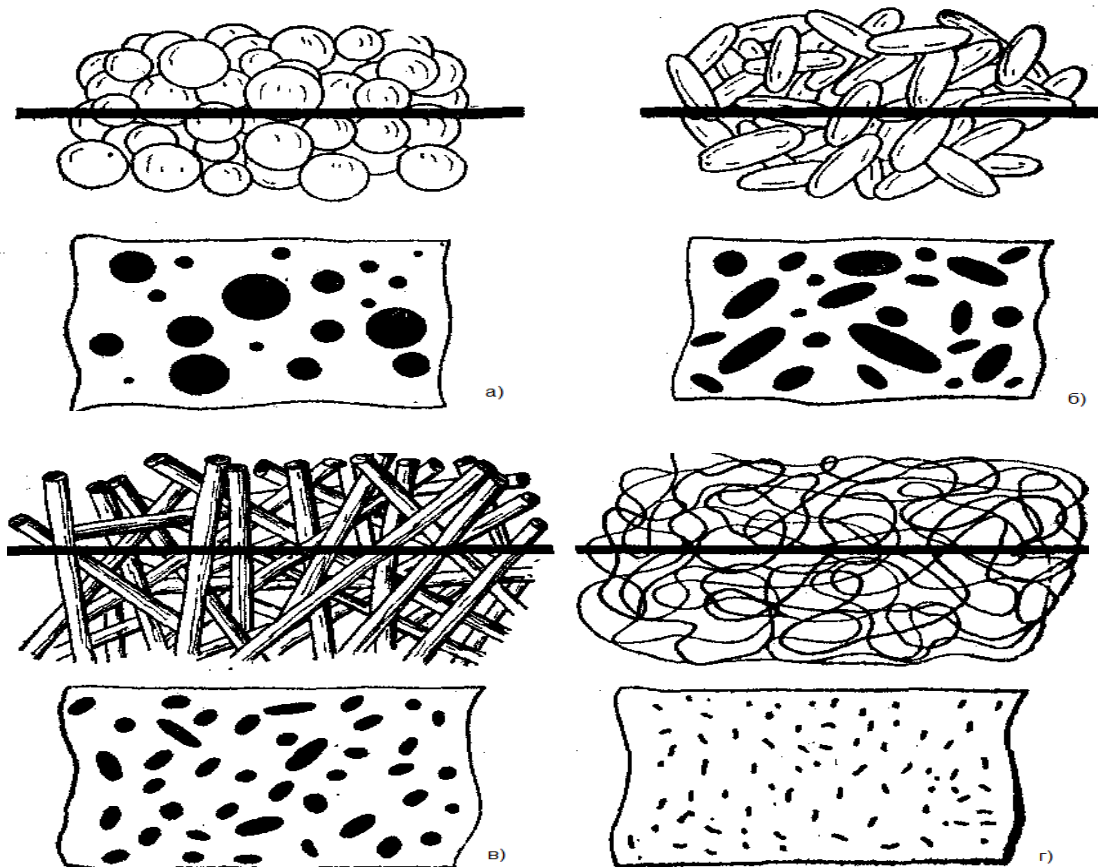
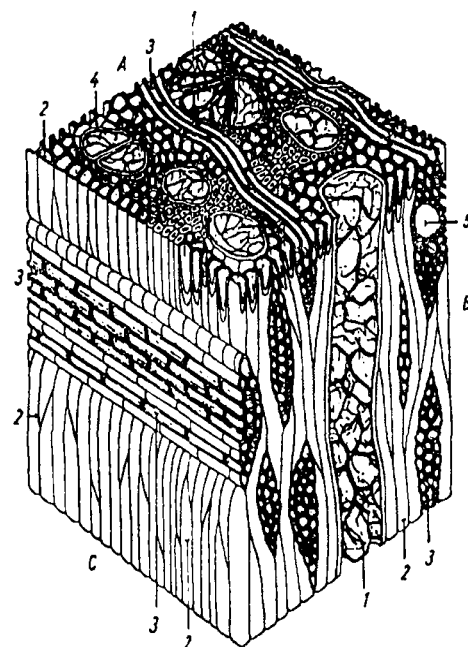


Рис. 1. Модели объектов, составленных из однородных структурных элементов и их сечения



А - поперечный срез;
 В - тангентальный срез;
 С - радиальный срез;
 1 - сосуды с тиллами;
 2 - волокнистые
 элементы;
 3 - сердцевинные лучи;
 4 - осевая паренхима;
 5 - межклетный ход в
 луче

Рис.2. Схема микроскопического строения лиственной древесины (по Wagenfur - у)

Числа Бетти.

Правила построения структур задаются физико–химическими законами взаимодействия молекул. В геометрических преобразованиях, происходящих в биологических объектах можно выделить две стадии: изменение кривизны их поверхности и увеличение количества связанных фигур. Средняя кривизна связана с внешней формой поверхности и пропорциональна нормальному давлению, искривляющую пленку, что компенсируется силами поверхностного натяжения. Разрывы и разрезание фигур нарушает их топологическую эквивалентность. Для описания свойств топологических преобразований могут быть использованы числа Бетти:

$$p_1 = b - n + p_0,$$

где p_0 - нулевое число Бетти, p_1 - первое число Бетти, b – число связей, n – число узлов (рис. 3).

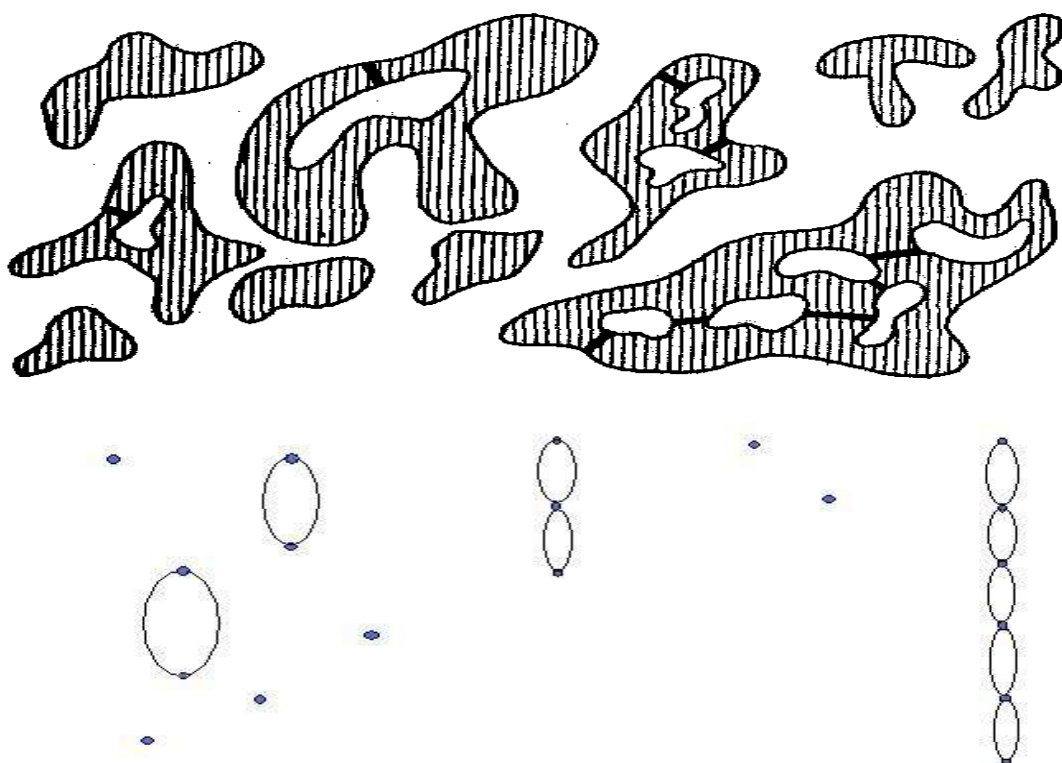


Рис.3. Пояснение физического смысла чисел Бетти, ряд ограниченных двумерных областей со сложными связями и их представление в виде графа: $p_0=10$, $p_1=9$, число связей $b = 18$, число узлов $n = 19$.

Кривизна

Кривизна k дуги плоской кривой является производной угла наклона касательной к положительному направлению оси абсцисс по длине дуги.

$$k = \frac{d\alpha}{dl}, \quad \text{где } \alpha \text{ – угол наклона касательной к оси абсцисс, } l \text{ –}$$

длина дуги.

Кривизна K поверхности

$$K = \frac{d\Omega}{dS}, \text{ где } \Omega - \text{ телесный угол, } S - \text{ площадь поверхности.}$$

Средняя кривизна $K_{cp} = \frac{\Omega_v}{S_v}$, где Ω_v – удельный сферический прообраз, приведенный к единице объема, S_v – удельная поверхность сегмента на единицу объема.

При анализе внутренней структуры трехмерных объектов сложной конфигурации используют удельные числа Бетти, отнесенными к единице объема:

$$K_{cp} = 4\pi \frac{(P_{0v} - P_{1v})}{S_v}$$

Аналогичное выражение для средней кривизны кривой на плоскости:

$$k_{cp} = 2\pi \frac{(P_0 - P_1)}{l_{cp}}, \text{ где } l_{cp} - \text{ длина грани-$$

цы объекта на плоскости.

Пример: Пусть внутри круглого объекта радиуса $R_1 = 4$ находятся три пустоты круглой формы радиуса $R_2 = 2, R_3 = 1, R_4 = 1$ (рис.4.).

Кривизна каждой окружности составляющее общую границу объекта будет:

$$k_1 = 1/R_1 = 1/4, \quad k_2 = -1/2, \quad k_3 = -1, \quad k_4 = -1.$$

Определим общую границу объекта:

$$L = 2\pi (4+2+1+1) = 16\pi.$$

Числа Бетти:

$$p_0 = 1, \quad p_1 = 3.$$

Вычислим среднюю кривизну общей границы:

$$k_{cp} = 2\pi(1 - 3)/16\pi = -1/4.$$

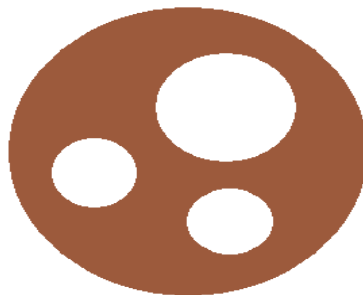


Рис.4. Объект круглой формы с тремя кругами (пустотами) внутри.

Реализация алгоритма определения чисел Бетти.

Программа по определению чисел Бетти основана на алгоритме определения связности графических объектов [3], при этом алгоритм работает в два этапа:

за первый проход программа определяет нулевое (p_0) число Бетти, делает оценку площадям и границам ограниченных двумерных областей со сложными связями (S_i и L_i), вычисляет суммарную площадь и суммарную оценку границы всех ограниченных областей ($S_{об}$ и $L_{об}$);

за второй проход программа определяет первое (p_1) число Бетти и делает оценку средней кривизне (k_{cp}).

Площади и длины границ в изображениях зависят от масштаба увеличения регистрирующих приборов (микроскопов), а так же от преобразований графических редакторов. Перечень схематических изображений признаков и их микрофотографии в древесине, используемых в данной работе, представлены М. И. Колосовой (кафедра общей экологии, физиологии растений и древесиноведения). Демонстрация работы программы представлена на (рис. 5. по рис. 12).

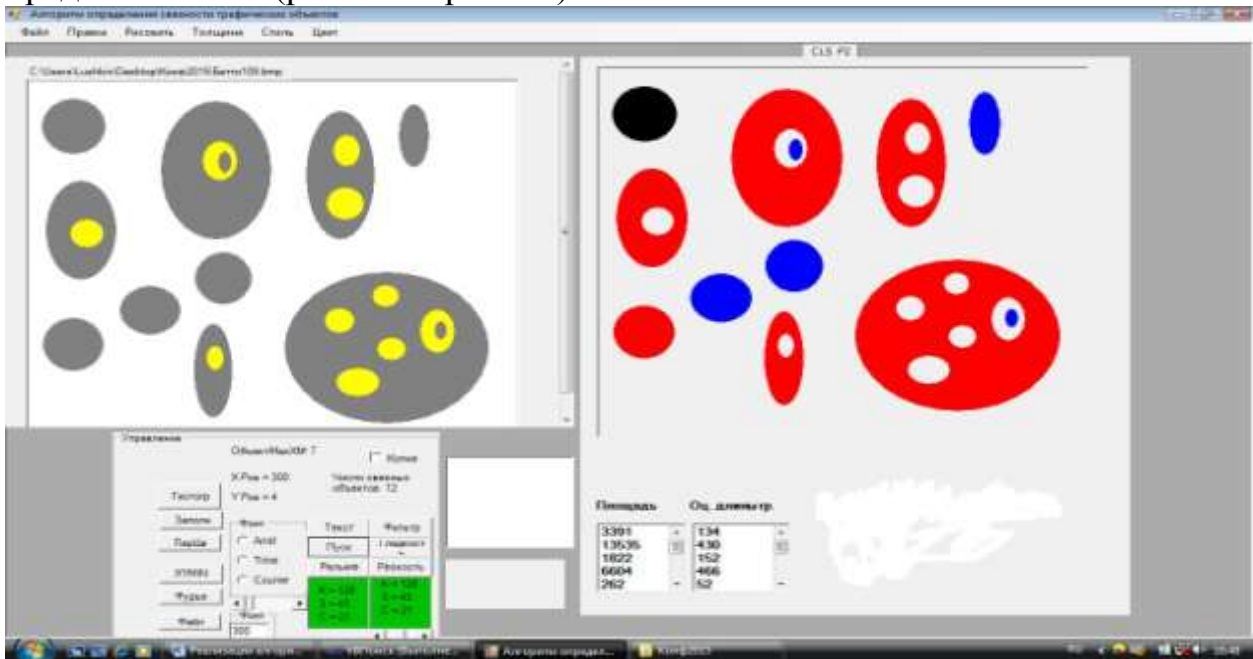


Рис.5. Демонстрация работы первого прохода программы число связных объектов $p_0 = 12$.

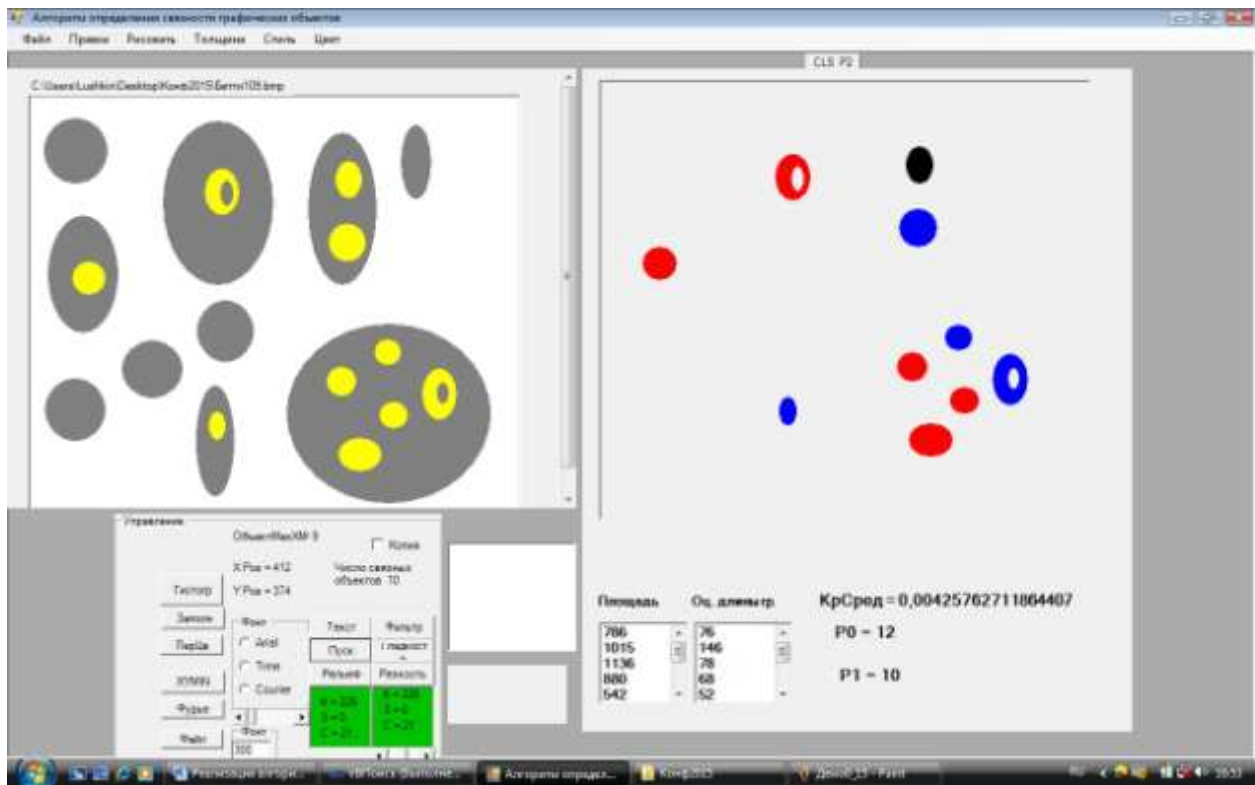


Рис.6. Демонстрация работы второго прохода программы $p_0 = 12$, $p_1 = 10$, оценка средней кривизны $k_{cp} = 0.004$

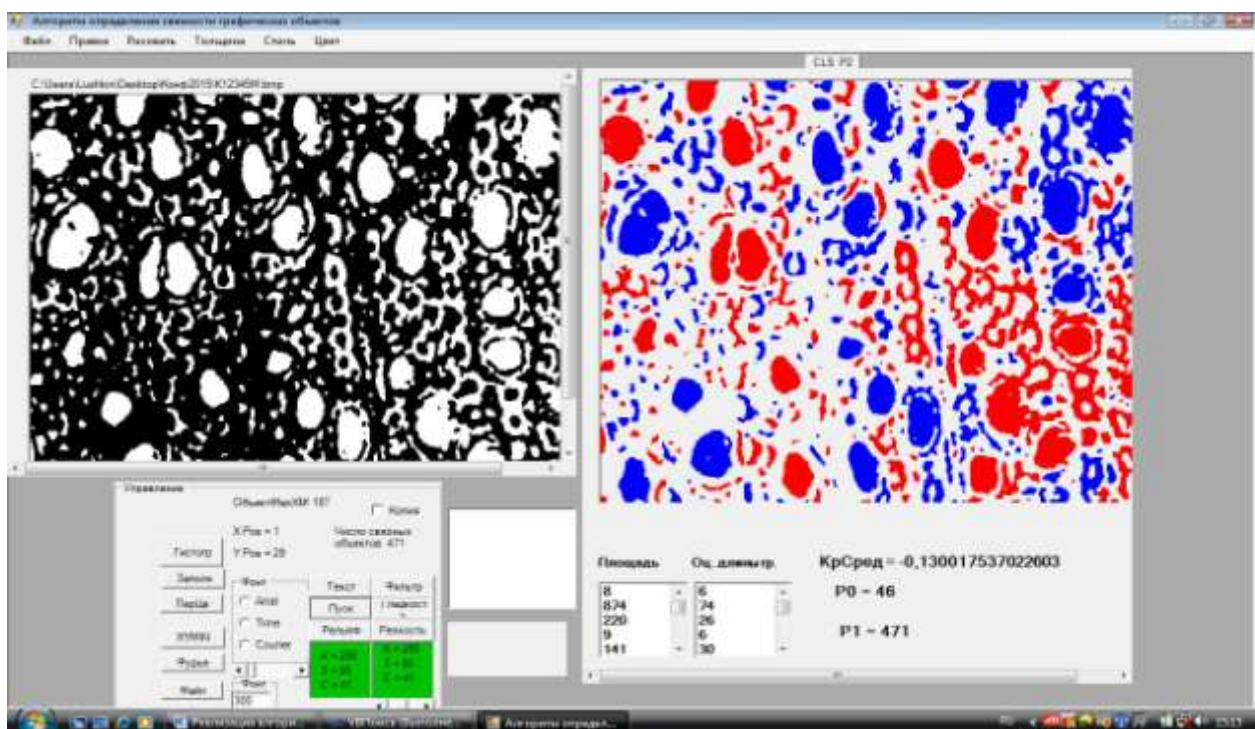


Рис.7. $p_0 = 46$, $p_1 = 471$, оценка средней кривизны $k_{cp} = -0.13$ (Бесконтактные сосуды в древесине самшита)

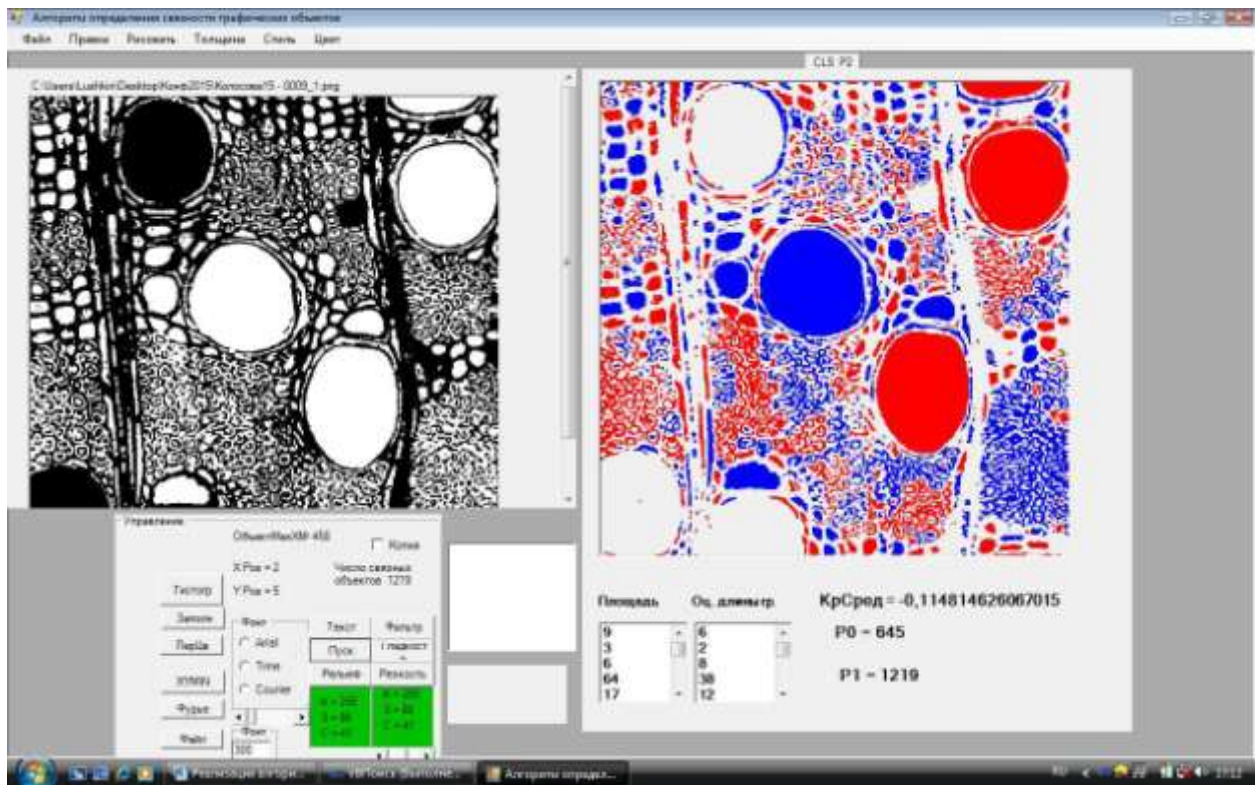


Рис.8. $p_0 = 645$, $p_1 = 1219$, оценка средней кривизны $k_{cp} = -0.115$
(Сосуды в древесине *Entandrophragma cylindricum* Sprague)

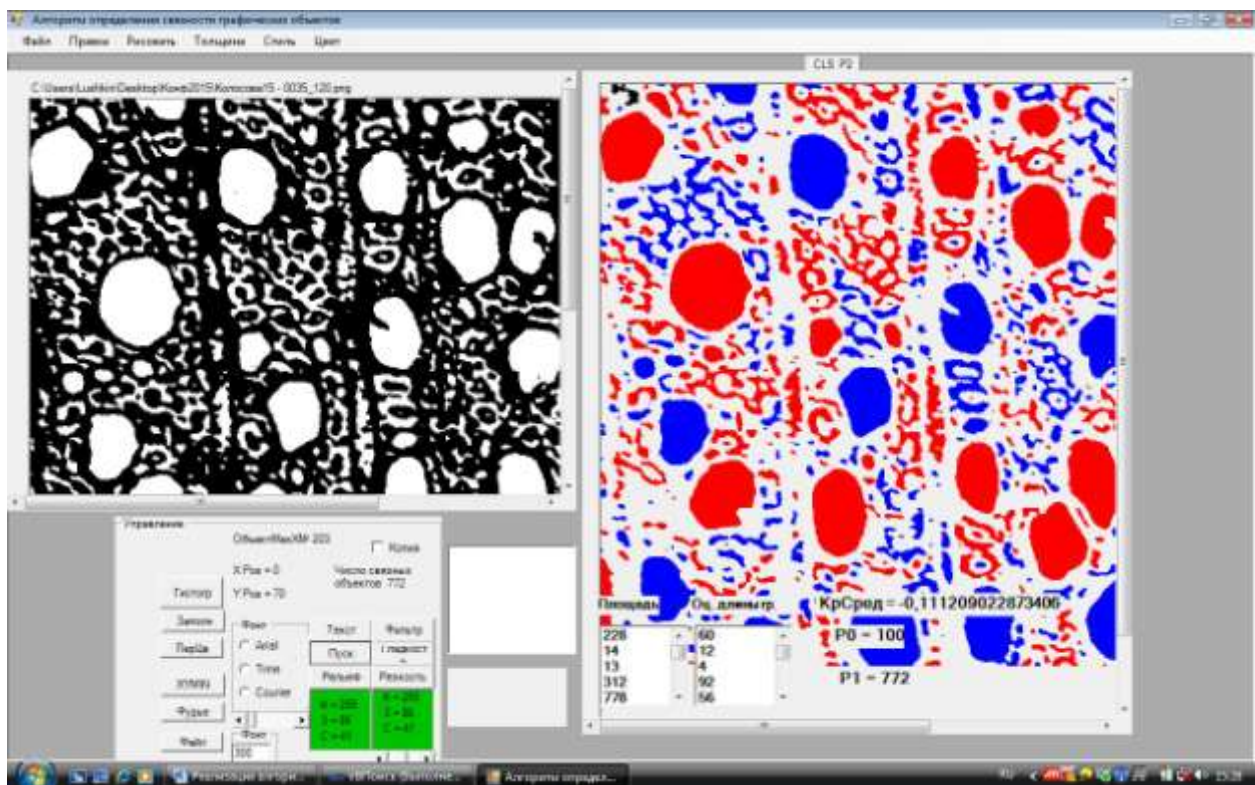


Рис.9. $p_0 = 100$, $p_1 = 772$, оценка средней кривизны $k_{cp} = -0.111$
(Бессосудистая зона в позднем слое прироста древесины груши)

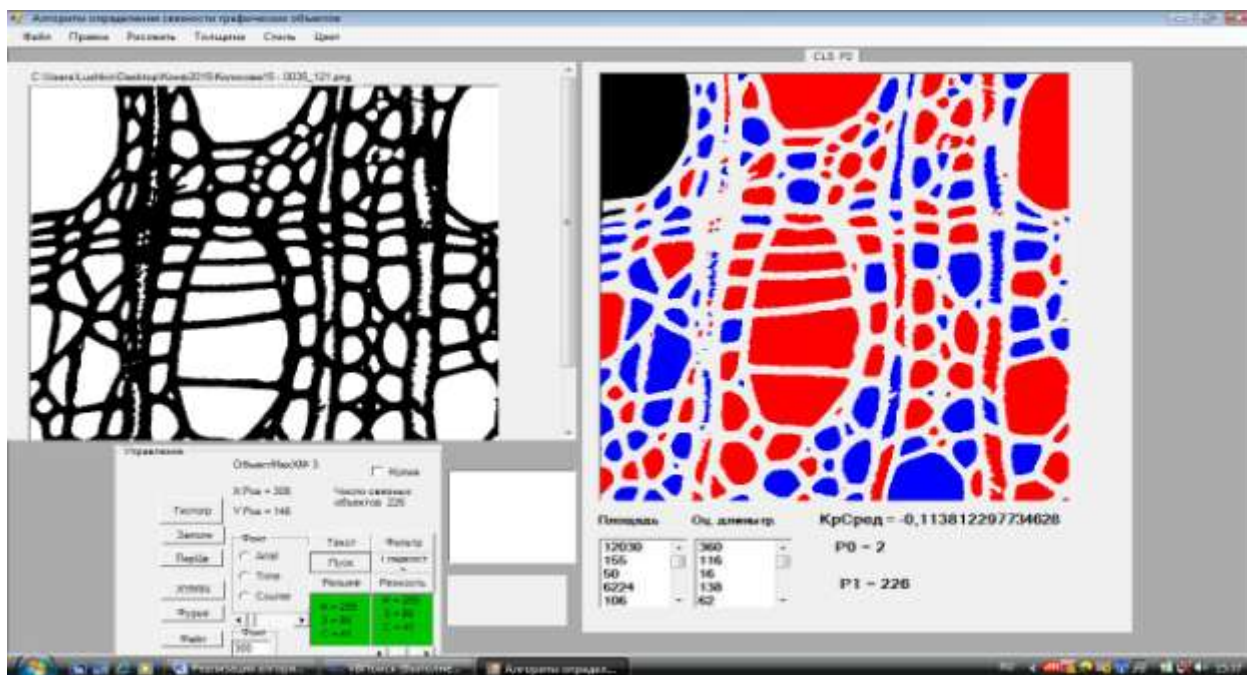


Рис.10. $p_0 = 2$, $p_1 = 226$, оценка средней кривизны $k_{cp} = -0.114$
 (Уменьшение поперечного сечения сосудов в цепочной группе на границе слоя в древесине ивы)

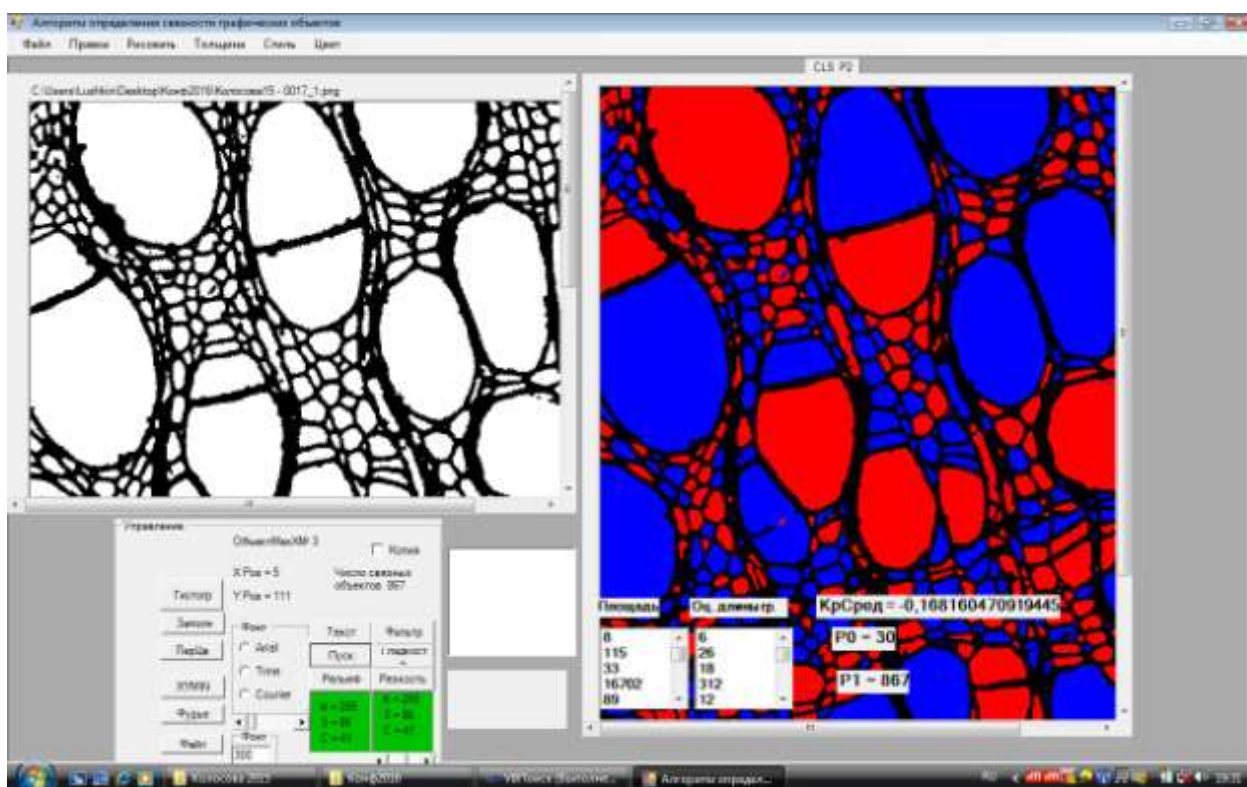


Рис.11. $p_0 = 30$, $p_1 = 867$, оценка средней кривизны $k_{cp} = -0.168$
 (Угловатые, разновеликие, тонкостенные волокнистые элементы в древесине ивы)

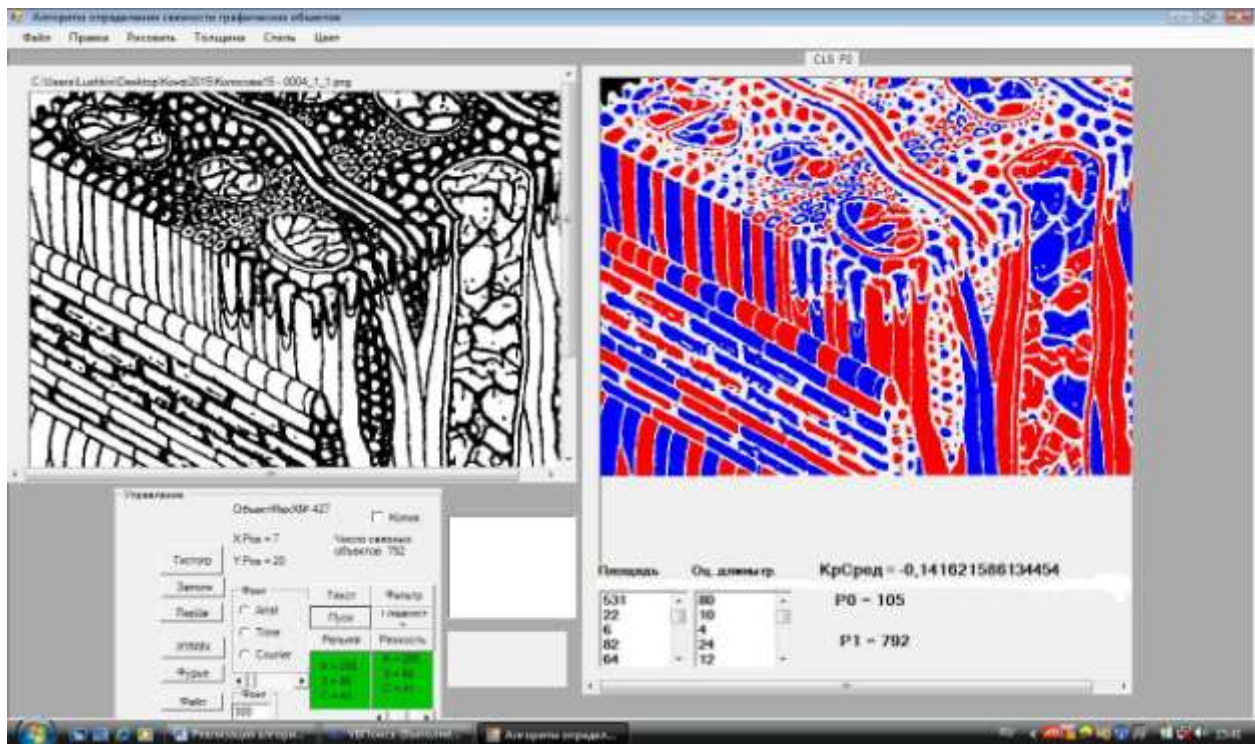


Рис.12. $p_0 = 105$, $p_1 = 792$, оценка средней кривизны $k_{cp} = -0.141$
(Схема микроскопического строения лиственной древесины)

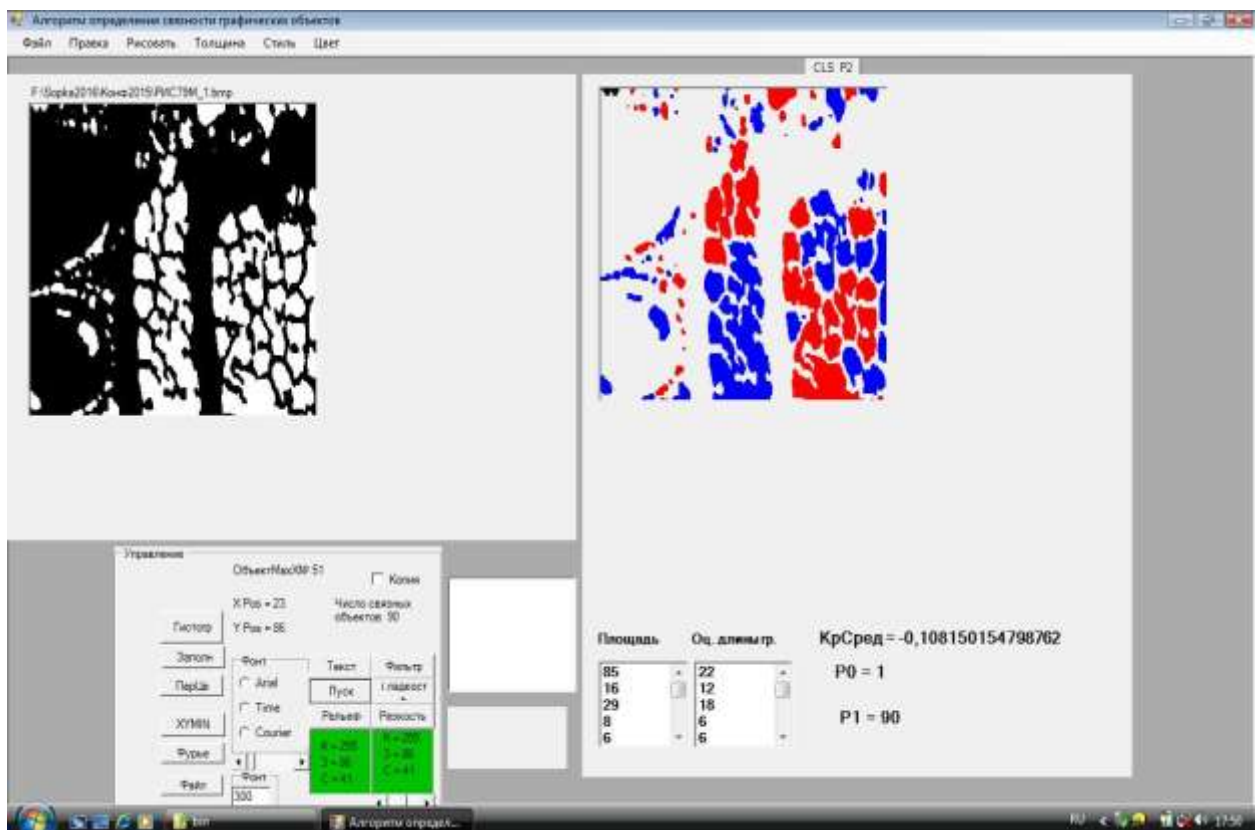


Рис.13. $p_0 = 1$, $p_1 = 90$, оценка средней кривизны $k_{cp} = -0.108$
(Поперечный срез древесины Manilkara sp)

Заключение

Заявленный подход при обработке графических файлов, позволяет выявить актуальные направления и обратить внимание на развитие компьютерного анализа объектов лесной отрасли.

Программная реализация алгоритма позволяет создать систему оперативной обработки микрофотографий древесины и оценки их анатомических признаков.

Библиографический список

1. Иваницкий Г. Р., Куниский А. С. Математические методы исследования структур. М., «Знание», 1975.
2. Колосова М. И., Соловьева Н. Г. Основные анатомические признаки древесины лиственных деревьев и кустарников. СПбГЛТУ, 2013.
3. Лушкин Н.В. Алгоритм определения связности графических объектов. - Труды Санкт-Петербургской Государственной лесотехнической академии Актуальные проблемы развития высшей школы Санкт-Петербург. 2010. Стр. 308-309.

М. О. Лебедев, кандидат технических наук, доцент

СОЗДАНИЕ КЛИЕНТСКИХ ПРИЛОЖЕНИЙ ДЛЯ ИНФОРМАЦИОННЫХ СИСТЕМ СО СЛОЖНОЙ СТРУКТУРОЙ ДАНЫХ

Рассматриваются общие принципы создания приложений, как клиент-серверных, так и однопользовательских (настольных или desktop'ных), работающих с динамическими информационно-поисковыми и регистрирующими системами (ИПРС). По динамическими ИПРС понимаются такие базы данных (БД), для которых состав фиксируемых в параметров не определен полностью на момент проектирования системы, или этот состав может меняться в ходе эксплуатации системы. Обобщенная структура ИПРС на логическом уровне в большинстве случаев может быть представлена схемой, показанной на рис. 1.

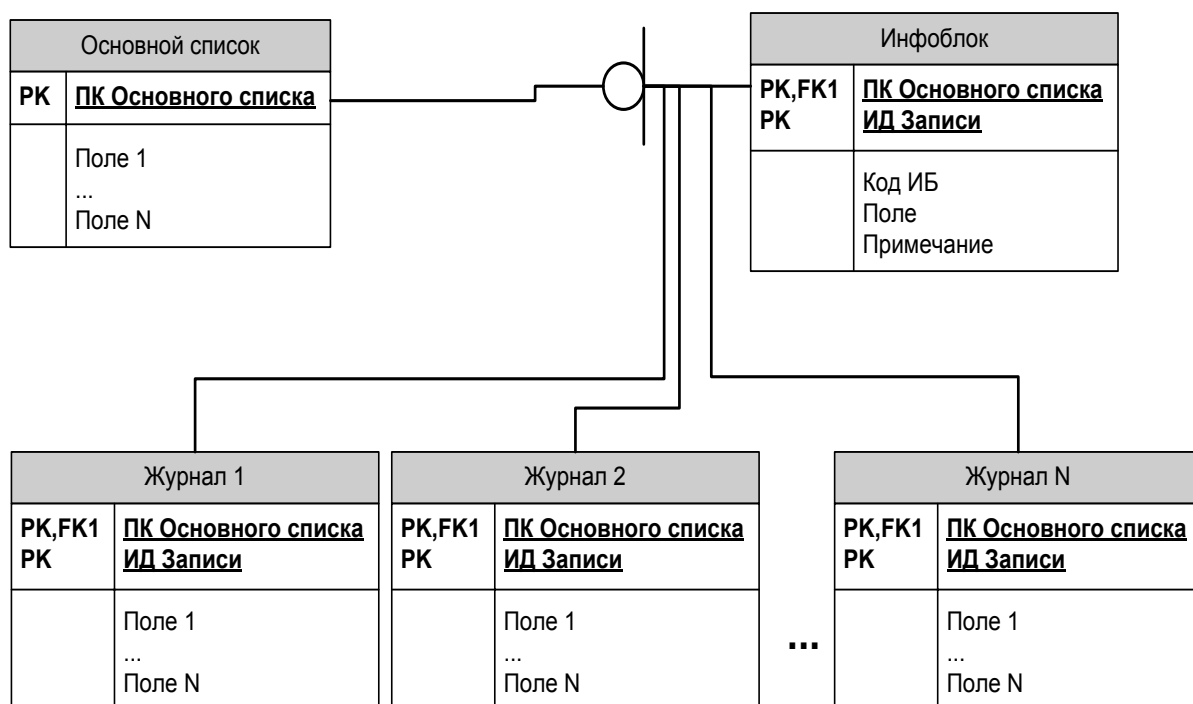


Рис. 1. Обобщенная схема ИПРС

Таблица "Основной список" содержит список основных объектов, для работы с которыми используется данная система. Это может быть список физических лиц (учащихся, пациентов, сотрудников и т.д.), список строений, список изделий и т.п. Каждый из этих списков будет иметь свой собственный состав полей, характеризующий объект, неразрывно с ним связанный, и присутствующий у любого элемента (объекта) данного списка. Так, например, если это список физических лиц, то у каждого объекта этого списка должны быть поля "Фамилия", "Имя", "Дата рождения", "Пол". Причем, состав этих полей не изменится будет ли это список учащихся какого-либо образовательного учреждения, или список сотрудников какой-либо организации. Набор данных "Инфоблок" содержит информацию, также характеризующий каждый объект списка. Но состав этой информации будет различаться от назначения информационной системы. Например, если это список учащихся, то в состав этого блока может входить, например, адреса регистрации и фактического проживания, информация о родителях, характеристика семьи. Для списка пациентов такой блок может содержать информацию об отделении и палате, куда помещен пациент, наличии аллергических реакций и т.п. Общим, не зависящим от назначения информационной системы, для этих блоков является то, что представленная в них информация является статической или квазистатической (редко изменяющейся). При изменении такой информации нет необходимости запоминать прежние значения измененных полей. По-сути дела "Инфоблок" представляет расширенную актуальную информацию об объекте.

Список наборов данных "Журналы" включает списки событий, связанных с объектом, так или иначе организованных в хронологическом поряд-

ке. При этом хронологическая организация может как включать в состав полей этих наборов данных дату и время, так и просто регистрировать события под очередным порядковым номером. Очевидно, что состав полей этих наборов данных различен не только между информационными системами различного назначения, но и внутри одной и той же ИПСР.

Не редко состав полей информационного блока (набора данных "Инфоблок") настолько объемный, что пользователю бывает затруднительно воспринимать представленную в нем информацию. В этом случае целесообразно группировать такую информацию по каким-либо признакам. Например, блок информации "О родителях" (если ИПРС связана с регистрацией учащихся), или "Перенесенные заболевания" (если информационная система о пациентах). Именно с этой целью в набор данных "Инфоблок" внесено поле "Код ИБ" (код информационного блока). Физически и набор данных "Инфоблок", и список журналов обычно представляют собой систему таблиц из операционных и справочных, имеющих отношения "один ко многим". Такая система характерна для классических хорошо нормализованных БД, исключая (или сводящих к минимуму) дублирование информации в информационной системе. Изменение состава фиксируемых в классической информационной системе параметров (полей) влечет изменение структуры БД (как изменение структуры отдельных таблиц, так и, возможно, изменение состава таблиц БД). В свою очередь, изменения структуры БД требуют изменений – перепрограммирования – как сервера БД, так и клиентского приложения. Использование динамических ИПРС позволяет свести к минимуму (а в идеальном случае, полностью исключить) работы по перепрограммированию приложений, работающих с информационной системой. Это достигается за счет того, что нормализация таблиц такой БД выполняется не на уровне характеристик объектов, а на уровне типов данных, описывающих эти характеристики. Т.е., структура таблиц и информационного блока, и журналов может быть представлена следующим образом (рис. 2).

Журнал I	
РК	<u>ПК Основного списка</u>
РК	<u>Код журнала</u>
РК	<u>Код поля</u>
РК	<u>Код записи</u>
	Значение

Рис. 2. Структура информационных таблиц динамической ИПРС.

Здесь, "Журнал I" - таблица, хранящая данные i -го типа (i - строковый тип, целочисленный тип, вещественный тип, логический тип и т.д.); "Код_поля" - определяет логическое имя поля, его тип данных, и способ ввода значения (вручную или выбирая значение из списка - из справочни-

ка); "Код_журнала" - идентификатор журнала на логическом уровне (см. рис. 1), в котором должны находиться эти данные (на логическом уровне); "Код_записи" - поле, однозначно определяющее запись в логических журналах или очередное значение параметра в информационном блоке (если параметр на логическом уровне допускает множественность значений); "ПК Основного списка" - поле (или состав полей), определяющий к какому объекту основного списка относится это значение. В таблицах такой структуры хранятся данные, которые на логическом уровне могут быть представленные как в журналах, так и в информационном блоке. Локализация данных (принадлежность к конкретному журналу/информационному блоку) определяется полем "Код_журнала".

Для создания приложения, работающего с такой структурой данных, необходимо иметь описание принадлежности полей к тому, или иному журналу. Это вопрос решается с помощью создания таблиц, описывающих варианты журналов (динамический журнал, или квазистатический журнал - информационный блок), и составы полей, относящиеся к логическим журналам. На рис. 3 показана структура таблиц, описывающих эти метаданные.

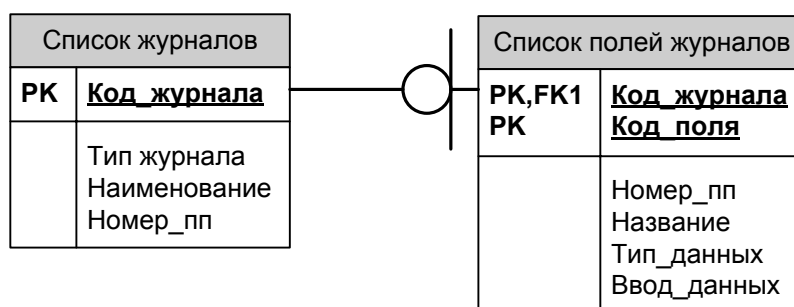


Рис. 3. Структура таблиц, определяющая состав журналов динамической ИПРС

Подобная схема организации данных не требует перепрограммирования ни сервера БД, ни клиентского приложения (для отображения, редактирования и поиска информации) при изменении состава параметров и журналов ИПРС. Однако, непосредственное использование такой структуры данных не отвечает всем требованиям современных информационных систем. В частности, достаточное количество полей в журналах, с которыми непосредственно работает пользователь, являются полям просмотра (lookip-полями). Фактически это означает, что в журнале храниться не самое текстовое (или иное смысловое) значение, а код поля, однозначно определяющий это значение в некой справочной таблице. Для обеспечения отображения смыслового значения (понятного пользователю) при физическом хранении кода этого значения необходимо осуществить связь между операционной и справочной таблицами. При создании приложений для классической БД эта задача решается за счет создания переменных (в контексте визуального программирования - компонентов), обеспечивающих доступ к справочным таблицам, и настройке (установки определенных

свойств) поля операционной таблицы, которое должно отображать это смысловое значение. В современных системах программирования (Visual Studio, Borland/Embarcadero RAD и т.п.) такую настройку можно выполнить как на этапе разработки приложения, так и на этапе выполнения программы. Но и в том, и в другом случае требуется заранее знать и имя переменной (компонента) справочного набора данных, и имена ключевых полей и lookup-поля. Для динамических ИПРС такая информация не может быть получена на этапе разработки приложения, она должна генерироваться автоматически на этапе выполнения программы. Критерием необходимости генерации такой информации является значение поля "Ввод_данных" (рис.3). Это поле (тип поля - любой перечислимый тип данных) содержит код, определяющий вводится ли значение вручную или выбирается из списка возможных значений. Для автоматической генерации lookup-полей в таблице "Список полей журналов" необходимо хранить свойства, необходимые для создания этих полей: "Хранимое поле" - поле, входящее в состав журнала, значение которого должно сохраняться при изменении lookup-поля (ключевое поле); "Отображаемые поля" - список полей, значения которых будут отображаться в клиентском приложении; "Источник данных" - SQL-запрос, определяющий набор данных, для выбора значения lookup-поля (при выборе какого-либо значения из этого списка изменяет значение поля, которое указано в свойстве "Хранимое поле"); "Отображаемые данные" - SQL-запрос, который возвращает значения отображаемых полей (полей, указанных в свойстве "Отображаемые поля") по значению ключевого поля (поля, которое указано в свойстве "Хранимое поле"). Такая структура позволяет создавать lookup-поля, источником данных которых являются не только справочные таблицы, но и другие журналы. Наличие этих полей в таблице "Список полей журналов" определяет необходимость создания соответствующих наборов данных при загрузке программы. Фактически, при загрузке приложения создается массив (или индексированный список) наборов данных, соответствующий журналам. Для каждого такого набора данных создается свой состав полей. Для полей, являющихся lookup-полями создаются массивы наборов данных для выбора значения и для отображения смыслового содержания ключевого lookup-поля.

В ходе разработки и эксплуатации ИПРС, построенных на основе данного подхода, выявилась необходимость связывать процесс редактирования данных журналов (добавление, удаление, корректировка) с какими-либо действиями над другими полями (имеется в виду логические поля) этих же журналов. Причем, такие поля не обязательно должны быть визуализированы. Например, при добавлении новой записи в журнал или корректировке существующей должны фиксироваться дата/время и оператор, выполнивший эту операцию. Очевидно, что описанный подход позволяет создать поля, фиксирующие эти параметры, но корректировка значений этих полей должна выполняться приложением, а не оператором. Фактиче-

ски, действия над такими полями выполняются при наступлении неких событий. Кроме того, для некоторых типов данных (в частности для строкового) выявилась необходимость создания их подмножеств. Например, в журнале появилась необходимость фиксировать неполную дату некоего события. Причем, требования по сортировке, фильтрации и выборке с учетом хронологии остались без изменения. Одним из возможных путей решения такой задачи является создание строкового поля, в котором дата сохраняется в формате ГГГГММДД. Где ГГГГ - символы, обозначающие год, соответственно, ММ - месяц, а ДД - день события. Если дата неполная, например, событие известно с точностью до месяца, на месте символов ДД указываются "00". Такой подход позволяет сохранять хронологическую выборку, используя условия "больше" - "меньше". Однако, для отображения даты в привычном для пользователя формате (ДД.ММ.ГГГГ) необходимо использовать специальные методы полей, позволяющие преобразовать физическое значение поля в его отображаемый вид. Здесь возникает две задачи:

- критерий того, что требуется преобразование физического значения поля в отображаемый вид (возможно, что будет необходимо и обратное преобразование - вводимого пользователем значения в необходимый формат);
- наличие метода, осуществляющего это преобразование.

Наиболее просто эта задача решается за счет создания подмножества типов полей. В список типов данных (поле "Тип_данных", рис.3) добавляется новый тип. Этот новый тип на физическом уровне остается уже существующим в списке типом (в рассмотренном примере это строковый тип данных), и, соответственно, все значения этого поля будут храниться в той же таблице. А в приложении для этого типа данных должны быть созданы методы преобразования физического значения поля в отображаемое и обратно.

Записи в динамических журналах (логические записи - записи в классическом смысле) представляют собой совокупности полей различных типов данных. При хранении значений одной записи в совокупности таблиц (в соответствии с типом данных ее полей) для отображения всей логической записи необходимо использовать SQL-запросы с объединением нескольких таблиц. Такой SQL-запрос может быть создан динамически в процессе выполнения программы по условиям вхождения физических записей из таблиц "Журнал I" (рис. 2) в ту или иную запись (для чего используется поле "Код_записи") Порядок, в котором должны быть видны эти поля, определяется значением поля "Номер_пп" (таблица "Список полей журналов", рис.3). Очевидно, что при заполнении журналов операторами не все поля могут быть заполнены. Тем не менее, даже не полностью заполненная запись должна отображаться в журнале. Для отображения неполных записей может использоваться левое или правое внешнее объединение таблиц. Однако, в этом случае необходимо, чтобы 1-я таблица (если

это левое внешнее объединение, или последняя, если это правое объединение) гарантированно содержало бы значение, относящееся к этой неполной записи. Так как гарантировать такое в условиях реальной эксплуатации информационной системы невозможно, то целесообразно использовать внутреннее объединение таблиц с обязательным заполнением всех полей записи. Т.е., при добавлении новой записи в журнал во всех таблицах "Журнал I" (рис. 2) автоматически добавляются записи, имеющие одинаковые значения полей "ПК Основного списка", "Код_журнала" и "Код_записи". В одной и той же таблице "Журнал I" для одной и той же совокупности перечисленных полей поле "Код_поля" может иметь различные значения. Это поле определяет колонку записи на логическом уровне. Значение поля сохраняется в поле "Значение" журнала. А, следовательно, если при заполнении записи это поле не было заполнено (или его значение было удалено), т.е. поле имело значение NULL, то необходимости в выборе каких-либо дополнительных правил для заполнения "пустых" полей специальными значениями. Такой подход сохраняет гибкость системы, возможность добавления новых полей в существующие журналы или добавление новых журналов. Негативной стороной является увеличение объема памяти БД. Порядок увеличения объема памяти рассмотрим на следующем примере. Пусть в журнале фиксируется дата обращения пациента, причина обращения и результат первичного осмотра. Причина обращения - частично формализовано, т.е., выбирается одно значение из некоего списка и вручную может быть дописано примечание. Классическая структура записи такого журнала может иметь вид, показанный ниже

ПК Основного списка	Код записи	Дата обращения	Код причины (из списка)	Примечание	Осмотр
---------------------	------------	----------------	-------------------------	------------	--------

Пусть поле "Код причины (из списка)" имеет целочисленный тип (первичный ключ этого справочника представлен целочисленным значением). Для представления этой же записи в динамической ИПРС потребуется SQL-запрос, объединяющий 4 набора данных (рис. 4).

Журнал Дата		Журнал Целое число		Журнал Текст - Примечание		Журнал Текст - Осмотр	
РК	<u>ПК Основного списка</u>	РК	<u>ПК Основного списка</u>	РК	<u>ПК Основного списка</u>	РК	<u>ПК Основного списка</u>
РК	<u>Код журнала</u>	РК	<u>Код журнала</u>	РК	<u>Код журнала</u>	РК	<u>Код журнала</u>
РК	<u>Код поля</u>	РК	<u>Код поля</u>	РК	<u>Код поля</u>	РК	<u>Код поля</u>
РК	<u>Код записи</u>	РК	<u>Код записи</u>	РК	<u>Код записи</u>	РК	<u>Код записи</u>
	Значение		Значение		Значение		Значение

Рис. 4. Состав таблиц, определяющих запись в динамической ИПРС

Две последние таблицы - это псевдонимы одной и той же таблицы, в которой хранятся текстовые данные. Как видно из данного примера для динамический ИПРС появляется необходимость хранить поля "Код_журнала" и "Код_поля", которые отсутствуют в классической структуре журнала. Кроме того, поля "ПК Основного списка" и "Код_записи" для данного примера занимают в четыре раза больше пространства, чем для классического журнала. Очевидно, чем больше полей (на логическом уровне) содержит журнал, тем больше памяти необходимо для хранения данных. Чрезмерное увеличение объема данных может приводить к замедлению работы приложения.

Альтернативой такой организации данных в сочетании с обеспечением динамичности ИПРС может быть создание журналов классической структуры на основе метаданных, описывающих эти журналы. При этом имена полей этих квазиклассических журналов могут формироваться по определенным правилам, включая в себя значения кода поля, указанного в метаданных при описании журналов. Как показала практика, использование такого подхода оправдано для динамических журналов, т.к. их объем информации существенно превышает объем статических (или квазистатических) журналов.

Данный подход был использован при разработке городской ИПРС "Профилактика правонарушений несовершеннолетних".

И. А. Обухова, кандидат технических наук, доцент
Р. М. Яковлев, кандидат физико-механических наук,
вед. научн. сотр.

РАДИАЦИОННАЯ БЕЗОПАСНОСТЬ В ОКОЛОЗЕМНОМ ПРОСТРАНСТВЕ И В КОСМОСЕ. РЕЗУЛЬТАТЫ РАСЧЕТОВ

Введение.

В предыдущей публикации [1] мы отметили, что дозы радиации, обусловленные космическим излучением за пределами Земли, многократно превышают установленную по НРБ-99/2009 предельную дозовую нагрузку и составляет 0,2 Зв в год (2 бэр в прежних обозначениях). Как нами отмечено в предыдущей публикации, предел дозового уровня составлял для космонавтов 150 бэр за год, затем было установлено (снижено) до 50 бэр (0,5 Зв). В этой работе мы покажем, что этот уровень будет намного

превышен при длительных полетах к Марсу и при пролете радиационного пояса Земли.

Космические аппараты и средства их доставки совершенствовались и сейчас уже время нахождения человека в них на орбите около Земли составляет от нескольких месяцев до года. Технические средства доставки человека на поверхность Марса в России и США также созданы. На орбиту Земли могут выводиться грузы более 30 т, а при сборке на орбите Земли из отдельных отсеков могут быть созданы космические аппараты весом в сотни тонн с необходимым запасом продовольствия и теплоносителя для дальних полетов (2-3 года) нескольких космонавтов к Марсу и обратно. Эти космические уже не аппараты, а корабли могут быть дополнены средствами доставки нескольких космонавтов на поверхность Красной планеты.

Но при длительном межпланетном полёте, а также при нахождении на поверхности Луны или Марса остаётся нерешенной проблема радиационной безопасности, обусловленная постоянно присутствующим в космосе галактическим космическим излучением (ГКИ) и солнечным космическим (СКИ) во время вспышек на Солнце. Высокий уровень радиации, создаваемый галактическим и солнечным корпускулярными излучениями, ограничивает пребывание людей в космосе. Усредненные состав и спектры космических лучей, приведены на рис.1-2.

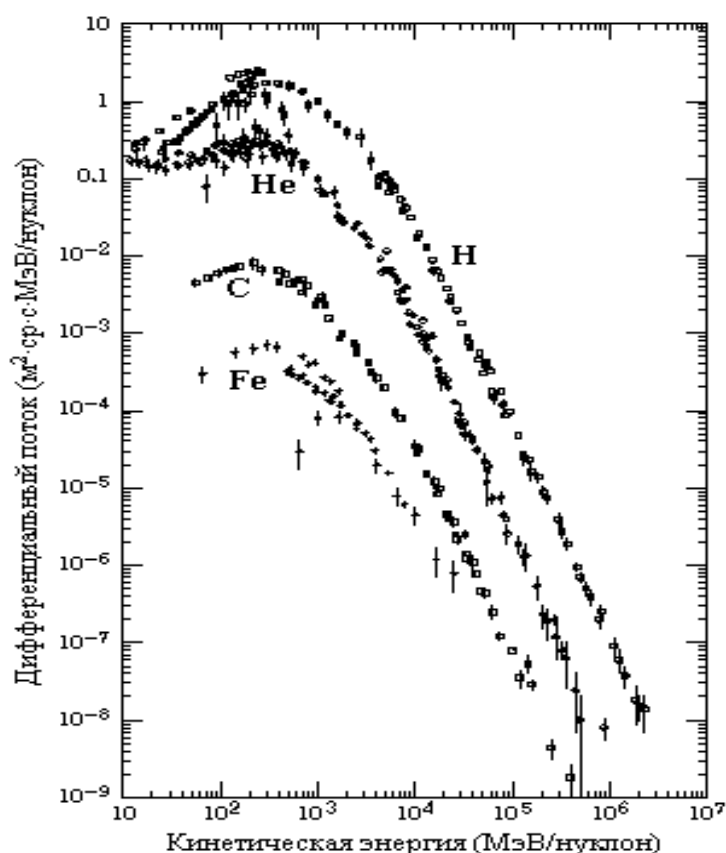


Рис. 1. Основные компоненты первичных космических лучей

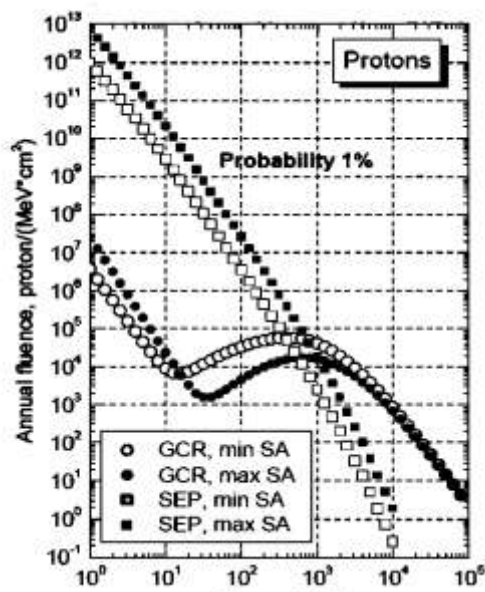


Рис. 2. Спектры космических лучей

При прохождении первичных космических лучей с высокими энергиями создаётся большое число вторичных частиц – пионов, нейтронов, протонов и более тяжелых атомных ядер. Вместо одной первичной частицы возникает большое число вторичных частиц. В одном акте взаимодействия протон обычно теряет ~50% своей энергии и в результате взаимодействия рождаются в основном пионы, а из ядра мишени вылетают нейтроны, однозарядные, двухзарядные и более тяжелые фрагменты с энергиями до 20 МэВ, линейные потери энергии которых велики. Последующие взаимодействия первичной частицы и пионов с ядрами приводят к дополнительному образованию сильноионизирующих частиц. Повреждения в биологической ткани зависят от поглощённой энергии излучения в единице объёма, а также линейными потерями энергии. Величиной этих характеристик определяются значения поглощенной и эквивалентной доз. При прохождении частиц высокой энергии через вещество за счёт ядерных взаимодействий эти дозы оказываются значительно выше значений доз без учёта ядерных взаимодействий. Поэтому при оценке радиационной ситуации в космосе эти взаимодействия необходимо правильно учитывать, но до этого характеристики этих взаимодействий надо было получить.

Первые характеристики неупругих взаимодействий частиц высоких энергий с атомными ядрами были получены ещё до создания ускорителей частиц высоких энергий при запуске с помощью шаров зондов стопок с многими слоями ядерных фотоэмульсий и запуске электронной аппаратуры. Эти исследования в СССР выполнялись в Москве под руководством Д.В.Скобельцина и С.Н.Вернова, а в Ленинграде в Радиевом институте им. В.Г.Хлопина в лаборатории Космических Лучей проф. А.П. Жданова, получившего Сталинскую премию вместе с С.Н.Верновым за эти работы.

После запуска ускорителей в работы в этом направлении были продолжены в Объединенном институте Ядерных Исследований (г.Дубна) на ускорителях Лаборатории Ядерных Проблем (протоны до 660 МэВ) и Лаборатории Ядерных Исследований (протоны и другие ядра с энергией от 1 до 10 ГэВ на нуклон). От Радиевого института во всех работах организатором и основным исполнителем работ был Р.М.Яковлев. Работы велись по договорам с ГЕОХИ им. В.И. Вернадского, Институтом Космических Исследований (ИКИ) и Институтом Медико-Биологических Проблем (ИМБП) и представлены в многочисленных отчетах Радиевого института перечисленным выше организациям. Основные результаты были опубликованы в совместных работах с сотрудниками перечисленных выше организаций и докладывались на различных совещаниях и конференциях. В этой статье мы приводим ссылки на часть тех работ, выполненных сотрудниками РИ и ИМБП, которые вместе с другими работами легли в основу норм радиационной безопасности НРБ-76-87 и представлены в таблицах этих норм.

До обсуждения радиационной ситуации в космосе рассмотрим сначала существующие нормы радиационной безопасности (НРБ) для работников атомной отрасли и для остальных жителей на Земле. Эти нормы основаны на многолетних исследованиях разрушительного действия радиации на клетки и отдельные структуры живого организма, а также на всестороннем изучении дальних последствий облучения не только для самих облученных организмов, но и для их потомства в нескольких поколениях.

Результатом этих исследований явились рекомендации радиационной безопасности, которые изложены в документах Международного Комитета по Радиационной защите (МКРЗ-60, МКРЗ-102 и др.). В различных странах учитывают эти рекомендации при создании и корректировке национальных норм радиационной безопасности (НРБ). Действующими сейчас в России являются НРБ-99/2009, они приняты на государственном уровне и все положения этого документа обязательны к исполнению. Согласно НРБ-99/2009 основные пределы доз составляют для персонала **20 мЗв (2 бэр в год) в среднем за любые последовательные 5 лет, но не более 50 мЗв в год**, а для населения **1 мЗв в год в среднем за любые последовательные 5 лет, но не более 5 мЗв в год**. Пределы доз в случае аварийных ситуаций могут быть превышены только в случае необходимости спасения людей и предотвращения их облучения. В соответствии с п. 3.2.3. НРБ - Лица, подвергшиеся облучению в эффективной дозе, превышающей 100 мЗв в течение года, при дальнейшей работе не должны подвергаться облучению в дозе свыше 20 мЗв за год. Облучение эффективной дозой свыше 200 мЗв в течение года должно рассматриваться как потенциально опасное. Лица, подвергшиеся такому облучению, должны немедленно выводиться из зоны облучения и направляться на медицинское обследование. Последующая работа с источниками излучения этим лицам может быть разрешена только в индивидуальном порядке с учетом их согласия по ре-

шению компетентной медицинской комиссии. Причём, планируемое облучение в эффективной дозе до 200 мЗв в год допускается только федеральными органами исполнительной власти, уполномоченными осуществлять государственный санитарно-эпидемиологический надзор.

Из-за высокой интенсивности космических лучей и их высокой энергии (см.рис.2), получаемые космонавтами дозы оказываются намного превышающими предельные нормы в земных условиях. Пребывание человека в космосе должно быть запрещено, если исходить из этих норм. Поэтому были составлены ведомственные предельно допустимые уровни радиации, которые представлены в табл.2 [ГОСТ 25645.215-85. Нормы безопасности при продолжительности полета до 3 лет. М.: Госстандарт СССР,1986].

Сейчас они уменьшены в соответствии с ОКОП -2004 и составляют 50 бэр в год. Уровень при правильной оценке оказывается превышенным в несколько раз. Остановимся на безопасности в космосе. Космическое излучение является неустранимым фактором космического пространства. Величина получаемой дозы и уровень риска во много раз преуменьшались до начала этого века, уменьшаются они и сейчас. Обусловлено это в первую очередь тем, что трудно отказаться от мечты покорения космоса и посещения Луны и Марса Человеком.

На поверхности Земли защитой от космического излучения (солнечного и галактического) является магнитное поле, отклоняющее заряженные частицы с энергией ниже 1 ГэВ для околополярных широт и для экватора с энергией ниже 15 ГэВ. Но даже при этом на широте 52° , при геомагнитном обрезании 2 ГэВ, годовая доза на высоте 20 км составляет 12,5 мкЗв/час (11,5 Зв в год), т.е. почти в 6 раз превышает допустимую годовую дозу (2сЗв) работников атомной промышленности и в 160 раз (0,1 сЗв) дозу для населения. Но благодаря геомагнитному полю и защитному слою атмосферы (1000 г/см^2) интенсивность космического излучения, как видно из табл., снижается в 400 раз, и составляет величину 0,035 мкЗв/час (0,03сЗв в год) на поверхности Земли.

Т.о., при защите магнитным полем и слоем вещества (атмосферы) 1 кг/см^2 (10 тонн на 1 м^2) жизнь на Земле находится в безопасности при любых вспышках на Солнце и изменений интенсивности ГКИ.

Теперь от безопасной ситуации на поверхности Земли вернемся к верхним слоям атмосферы и в околоземное пространство (спутниковое), а затем рассмотрим радиационную ситуацию открытого космоса, к Луне и Марсу, где доза за защитным слоем атмосферы 55 г/см^2 .

В докладе [8] приводятся дозовые нагрузки, которые получает человек, значительно большие, чем в предыдущих публикациях ИМБП и более близкие к реальности.

Характеристики облучения человека в космическом полете

Источник излучения	Зона космического пространства	Поглощенная доза за сутки, Гр	Примечание
Галактические космические лучи	Орбита высотой 600 км	$1.2 \cdot 10^{-5}$ $I = 0^0$ $6.7 \cdot 10^{-5}$ $I = 90^0$	Среднетканевая поглощенная доза
	Межпланетное пространство	$130 \cdot 10^{-5}$ $260 \cdot 10^{-5}$	Максимум СА Минимум СА
	Поверхность Луны и Марса.	$70 \cdot 10^{-5}$ $140 \cdot 10^{-5}$	Максимум СА Минимум СА
Радиационные пояса Земли	На высоте 600 км	0.001 0.2	Протоны ср тк Электроны ср. тк.
	На высоте 1500-1800 км	До 10^4 До 10^6	Протоны ср тк Электроны пов.
Солнечные 23.02.1956 10.05.1959 03.09.1960 космические лучи 12.11.1960		2.8 54 0.02 10	Протоны. Поверхн. доза за вспышку

При оценке авторов доклада доза, полученная за неделю для космонавтов на орбите, параллельной оси вращения, составляет 20 мЗв, что соответствует *годовой дозе работника в атомной промышленности*. Т.е., годовая доза персонала реализуется для космонавта на такой орбите за неделю. За год она, соответственно, будет в 50 раз больше. При более низких орбитальных полетах (200-300 км) из-за малого количества протонов из радиационного пояса доза будет значительно меньше, но все равно оказывается в 5 - 6 раз выше дозы для профессионального работника атомной промышленности. При полете к планетам необходимо преодолеть радиационный пояс и как видно из таблицы при пролете радиационного пояса получать за день в 100 раз более высокую нагрузку, чем при орбитальном полете вблизи Земли. Доза составит не менее 1 Зв (при пролете зоны за 1 час). Дальнейшее длительное пребывание в космическом полете за год составит величину в минимум солнечной активности 1,186 Зв и в максимум солнечной активности 2,37 Зв только от галактического излучения. В случае вспышек на солнце эти величины могут быть на порядок выше. Итак, если пройти космическому кораблю радиационный пояс за 10 минут и далее находиться в открытом космосе 2 года (полет на Марс и обратно), то получится доза больше 200 бэр за год. Эта доза в 100 раз превышает дозу для жителей земли и в 4 раза предельно допустимую для космонавтов

(50 сЗв/год). Как это преодолеть? В работе [8] и в отчете МГУ [4] указывается, что увеличение защиты космического корабля может уменьшить величину радиационной нагрузки ниже 500 мЗв в год (50 бэр в год). Мы считаем, что увеличение защитного слоя не уменьшит значительно величину дозы из-за рождения нейтронов. На этом мы остановимся в следующем сообщении («Защита от космического излучения при длительных полетах»). В следующей работе мы более подробно остановимся на учете вторичных взаимодействий в защите и учтем роль нейтронов, которые определяют дозовую нагрузку за толщиной защиты более 50 см.

Как выше было отмечено, доза ниже за слоем атмосферы, при отсечении магнитным полем всех частиц с энергией ниже 2 ГэВ.

В межпланетном пространстве эта доза только за счет галактического излучения будет выше в 1,3 раза при учете облучения протонами и ядрами ГКИ, с энергиями ниже 2,5 ГэВ составит величину 40 бэр без учета альбедных нейтронов (мы их учтем в следующем сообщении) В открытом космосе доза составит около 80 бэр в год (еще в 2 раза больше) - за счет облучения аппарата без экранирования планетой со всех сторон. Солнечное излучение мы учтем ниже.

Как видно из [1], даже протоны при энергии 50 МэВ дают дозу 1 нбэр, хотя их пробег в алюминии 2-3 мм. Дозовую нагрузку создают и заряженные частицы меньших энергий. И для протонов радиационных поясов их надо учитывать. Учет этих частиц увеличивает дозовую нагрузку в 1,5-2 раза для солнечных вспышек и протонов радиационного пояса. Теперь, без учета радиационного пояса, за защитой 50 г/см² составит в открытом космосе доза составит до 20 бэр/год. Только увеличение защиты до 150 г/см² уменьшит дозовые нагрузки на порядок.

Таким образом, из-за образования (генерации) нейтронов увеличение защиты в разумных пределах не гарантирует безопасность, но может быть только при создании электромагнитного поля сравнительно высокой напряженности вокруг спутника позволит уменьшить радиационную нагрузку до приемлемых размеров. Но об этом только недавно начинают думать. С нашей точки зрения, пребывание человека длительное время в космических аппаратах и на поверхности Луны или Марса в открытом космосе очень некомфортно и без космического излучения, а космическое излучение создает смертельную опасность для здоровья.

Поэтому лучше отправлять в дальний космос и на планеты вместо людей исследовательские комплексы, включая взятием проб ими необходимого количества для исследования.

В докладе [8] не фиксируется, за какой защитой реализуется доза при орбитальных полетах. Но из этой таблицы видно, что без учета радиационных поясов величина годовой дозы от галактического космического излучения велика и составляет величину до 60 бэр/год. При этом эта доза реализуется за счет только высокоэнергетичной компоненты выше 2,5 ГэВ из-за геомагнитного обрезания.

При отсутствии магнитного поля в открытом космосе дозовая нагрузка в несколько раз выше. Кроме этого дополнительная доза возникнет за счет СКИ. В период максимальной солнечной активности доза, получаемая от СКИ, будет определяющей. Мы провели дополнительный учет дозовых нагрузок за счет части спектра ГКЛ ниже 2,5 ГэВ на нуклон и СКЛ в различные периоды солнечной активности при различных толщинах радиационной защиты от 10 до 50 г/см². При оценках мы использовали полученные ранее при участии одного из соавторов величины дозовых нагрузок при различных энергиях протонов, которые приводятся, взятые нами из [НРБ,1999] и экспериментальных данных по взаимодействию протонов с атомными ядрами для высоких энергий [5]. Надо подчеркнуть, что при взаимодействии протонов даже не высоких энергий ниже 200 МэВ, где пробеги частиц в веществе алюминий значительно меньше толщины защиты, получаемая от протонов доза составляет значительную величину. Ее необходимо учитывать. На рис.2 приводятся за защитой 50 г/см² от протонов энергий от 50 до 600 МэВ.

При этом частицы менее 200 МэВ имеют пробег несколько сантиметров, см.табл.2.

Т а б л и ц а 2

Пробеги протонов в алюминии [17]

Энергия МэВ	5	10	20	40	100	1000
Пробег, см	$1,8 \cdot 10^{-2}$	$6,2 \cdot 10^{-2}$	$2,7 \cdot 10^{-1}$	$7,0 \cdot 10^{-1}$	3,6	148
Пробег, мг/см ²	50	170	560	$1,9 \cdot 10^3$	$9,8 \cdot 10^3$	$400 \cdot 10^3$

Т а б л и ц а 3

Эквивалентные дозы в бэрах от протонов за год. В скобках – значения доз от протонов с энергией более 2 ГэВ. при солнечном минимуме

Толщина защитного слоя	10	30	50	150
Доза от солнечного КИ (от 1 ГэВ)	8.14 (0.24)	4.37 (0.25)	3.23 (0.26)	3.19 (0.27)
Доза протонов ГКИ (от 1 ГэВ)	22.75 (16.0)	22.69 (17.65)	23.57 (18.9)	23.98 (16.22)
Доза от гал. и солн. протонов КИ (от 1 ГэВ)	30.89 (16.24)	28.06 (17.9)	26.80 (19.16)	27.17 (16.49)
От ядер ГКИ тяжелее протона	11.37 (8)	11.35 (8.83)	11.77 (9.45)	12 (8.11)
От всех активных частиц	42.26 (24.24)	39.41 (26.73)	38.57 (28.61)	39.17 (24.60)

Т а б л и ц а 4

Эквивалентные дозы в бэрах от протонов за год. В скобках – значения доз от протонов с энергией более 2 ГэВ. при солнечном максимуме

Толщина слоя в г/см ²	10	30	50	150
<i>От СКИ</i>	65.24	40.5	23.92	17.5
<i>От СКИ при энергиях более 1 ГэВ</i>	1.06	1.04	1.052	1.1
<i>От ГКИ</i>	11.85	9.6	10.15	10.46
<i>От ГКИ при энергиях более 1 ГэВ</i>	7.4	7.84	8.48	8.76
<i>СУММА от протонов СКИ и ГКИ при энергии более 1 ГэВ</i>	77.09 8.46	50.4 8.84	35.07 9.53	27.96 9.86

Т а б л и ц а 5

Значения эквивалентных доз от энергии протонов за защитой из алюминия

<i>Энергия протона, МэВ</i>	<i>Толщина экрана, г/см²</i>			
	10	30	50	150
30	0.2	0.15	0.1	0.05
75	5	2	0.6	0.3
200	40	20	10	5
500	130	120	120	100
850	135	140	145	130
2000	160	167	170	150
4000	180	200	220	220
6000	200	220	240	260
8500	220	240	270	290
20000	265	290	320	350
65000	322	345	384	420

Выводы

Приведенные нами данные показывают, что учет частиц низких энергий особенно для СКЛ составляет довольно высокую величину, ими нельзя пренебрегать, поскольку в СКЛ преобладают частицы низких энергий и их число весьма велико. Поэтому при длительных межпланетных полетах лучше создать искусственное магнитное поле на космическом корабле, обрезающее частицы низких энергий до 1 ГэВ. Такое магнитное поле позволит существенно снизить дозы радиации. В таком случае лучше лететь в период максимума солнечной активности, в противном случае предполагается полет при минимуме солнечной активности, что при длительном межпланетном полете даже с большой защитой приводит к слишком большим дозам облучения.

Библиографический список

1. Обухова И.А., Яковлев Р.М. Расчеты по космическим лучам и оценка радиационной опасности у земли и в космосе. Инф.системы и техн.:теория и практика.Сб.н.тр.Вып.7.СПб,СПбГЛТУ,2015.
- 2.Соболевский Н.М.. Актуальные вопросы радиационной безопасности длительных космических полетов. 2001 г., Дубна. Н.М.Соболевский.
- 3.Shaefer H.J. Galactic radiat.hazard in long-term space mission Aerspace Med.1968, v.39, N3, p.271-276, 1960,v31, p.807-816.
4. Панасюк М.И.НИР «Освоение СП-НИИЯФ» (этап 1 календ. плана договора от 20.10.2009г. № 118-09// 0901-1322/ 410-2009, п.п. 2.3.1 – 2.3.4, 3.1 – 3.7 ТЗ)/ МГУ.
5. Дудкин В.Е., Яковлев Р.М. и др. Нерелятивистские частицы, образующиеся при взаимодействии многозарядных ядер космического излучения с легкими эмульсиями.-«Яд.физика»,т.14,вып.2,с.378-381, 1973,т. 18,вып.2, с.374-377.
6. Дудкин В.Е., Яковлев Р.М. и др. Вторичные частицы, образующиеся при взаимодействии многозарядных релятивистских ядер с ядра на эмульсии «Яд.физика», 1973, т.18,с.374.
7. Dudkin V.E., Yakovlev R.M.and others. A Study of the Dose Composition in Tissue-Equivalent Phantoms for High-Energy Protons.-Health Physics, 1972,v.23, P.663
8. Петров В.Н. и др. Предложения по программе исследования ключ.фунд.пробл.обесп.рад.безоп.(косм.радиобиология). Докл. “Акт.вопр. рад. безоп. косм. полетов” .М., Дубна,2011.
9. Сакович В.А., Семенов В.Ф. Радиационная безопасность при пилотируемом полете на Марс. Концепция. Ат.энергия, т.99, вып.4, окт.2005, с.301-309.
10. Дудкин В.Е., Яковлев Р.М. и др. Изучение формирования доз в тканезквивалентных факторах. // В кн. Доклады совещания по дозиметрии и физике защиты на ускорителях. ОИЯИ 16-4888, Дубна, 1969, с.179-190.
11. Е. И. Воробьев, Ковалев Е.Е. Радиационная безопасность экипажей летательных аппаратов. М. Энергоатомиздат,1983. 151 с.
- 12.Григорьев Ю.Г. Радиационная безопасность космических полетов. М.,Атомиздат,1975.254 с.
13. Юревич В.И., Яковлев Р.М. и др. Образование нейтронов при взаимодействии протонов с энергией 2 ГэВ с ядрами. //ОИЯИ, Дубна, Радиевый инст-т. Ядерные инструм. и методы физ.исслед. А562 (2006) 747-749.
14. Юревич В.И., Яковлев Р.М. Исследование эмиссии нейтронов при взаимодействии релятивистских протонов и нейтронов. //ПисьмаЭЧАИ, т3,N2(132),2006

15. Дудкин В.Е. и др. // В сб. "Вопросы дозиметрии и защиты от излучений". Под ред. В.И.Иванова. Вып.3, М., Атомиздат, 1964, с.159.

16. Дудкин В.Е. и др. Методика расчета глубинных распределений доз от протонов высоких энергий за защитой. // Вопр. дозиметрии и защиты, вып.9. М., Атомиздат, 1969.

17. Физические величины. Под ред. И.С.Григорьева, Е.З.Мейлихова. М., Энергоатомиздат, 1991. 1232 с.

Н.Г. Полетаева, кандидат технических наук, доцент
Д.О. Туктарев, студент 4 курса

ЗАЩИТА ДАННЫХ КРИПТОГРАФИЧЕСКИМИ СРЕДСТВАМИ СУБД ORACLE

Классическая схема защиты баз данных (БД) подразделяется на следующие обязательные процедуры:

1. Разграничение доступа – каждый пользователь, включая администратора, имеет доступ только к необходимой ему согласно занимаемой должности информации.

2. Защита доступа – доступ к данным может получить пользователь, прошедший процедуру идентификации и аутентификации.

3. Шифрование данных – шифровать необходимо как передаваемые в сети данные для защиты от перехвата, так и данные, записываемые на носитель, для защиты от кражи носителя и несанкционированного просмотра/изменения не средствами системы управления БД (СУБД).

4. Аудит доступа к данным – действия с критичными данными должны протоколироваться. Доступ к протоколу не должны иметь пользователи, на которых он ведется.

В случае приложений, использующих многозвенную архитектуру, приведенные функции защиты также имеют место, за исключением защиты данных на носителе – эта функция остается за БД.

Всеми перечисленными функциями безопасности в той или иной мере оснащены СУБД и приложения Oracle, что выгодно отличает их от продуктов конкурентов.

Рассмотрим более подробно криптографические средства защиты информации в СУБД Oracle. Шифрование данных может повысить защиту как внутри, так и вне БД. Если один из столбцов БД зашифрован, и пользователь не знает ключа к этому шифру, информация перестает быть полезной. То же самое относится и к информации, которая должна быть без риска передана по сети.

Впервые появившийся в СУБД Oracle 10g пакет DBMS_CRYPTO заменяет инструментальное средство DBMS_OBFUSCATION_TOOLKIT (используется преимущественно с более ранними версиями). В него вклю-

чен алгоритм шифрования Advanced Encryption Standard (AES – усовершенствованный стандарт шифрования), заменивший Data Encryption Standard (DES – стандарт шифрования данных).

Пакет DBMS_CRYPTO имеет ряд преимуществ перед DBMS_OBFUSCATION_TOOLKIT:

- большой выбор алгоритмов шифрования, в частности, поддержка последнего усовершенствованного стандарта шифрования AES;

- возможность поточного шифрования, то есть организация потока предназначенных для шифрования данных;

- поддержка алгоритма защищенного хеширования SHA-1 (Secure Hash Algorithm 1);

- способность создания кода аутентификации сообщения MAC (Message Authentication Code);

- шифрование больших объектов (LOB) в их собственном формате.

Поскольку безопасность зашифрованных данных зависит от трудности угадывания ключа, выбор надлежащего ключа – самый главный шаг в процессе шифрования. Ключ может быть любым значением данных типа RAW, но если оно выбрано не достаточно случайно, злоумышленник будет в состоянии угадать ключ, это должно быть действительно случайное число. Случайные числа могут генерироваться с помощью встроенного пакета DBMS_RANDOM, но криптографически стойкая генерация случайных чисел достигается использованием функции RANDOMBYTES в пакете DBMS_CRYPTO. Функция имеет один входной параметр (типа данных BINARY_INTEGER), на выходе выдается число типа данных RAW, длина которого определена входным параметром. Это число может использоваться как ключ, что демонстрируется на примере простого PL/SQL-кода, приведенного в листинге 1.

Л и с т и н г 1. Шифрование данных

```
1  DECLARE
2  input_string VARCHAR2(255):='OMNIA MEA MECUM PORTO';
3  raw_input RAW(2000);
4  enc_val RAW(2000);
5  l_key RAW(2000);
6  l_key_len NUMBER:=128;
7  l_mod NUMBER:=dbms_crypto.ENCRYPT_AES128
8  + dbms_crypto.CHAIN_CBC+dbms_crypto.PAD_PKCS5;
9  BEGIN
10 raw_input:=
11 UTL_I18N.STRING_TO_RAW(input_string, 'AL32UTF8');
12 l_key := dbms_crypto.randombytes(l_key_len/8);
```



```

13  enc_val:= dbms_crypto.encrypt(
14  SRC => raw_input,
15  TYP => l_mod,
16  KEY => l_key);
17  --Здесь можно использовать зашифрованные данные enc_val
18  END;

```

Константа `l_mod` в строке 7 листинга 1 есть сумма трех констант `ENCRYPT_AES128` (алгоритм шифрования, размер ключа), `CHAIN_CBC` (размер блоков, на которые в процессе шифрования будет разбиваться исходная строка) и `PAD_PKCS5` (схема дополнения строки до требуемой кратности). Для проекта шифрования в пакете `DBMS_CRYPTO` подходит несколько типов алгоритмов и соответствующих длин ключей (табл. 1). Первый столбец в табл. 1 – имя константы – показывает константы, определенные в пакете для указания различных алгоритмов и длин ключей (Effective Key Length).

Надежность системы шифрования в целом зависит от того, как соединяются и зашифровываются блоки – по отдельности или совместно с соседними блоками. Oracle поддерживает методы сцепления, приведенные в табл. 2:

Т а б л и ц а 1

Типы шифрования в пакете `DBMS_CRYPTO`

Имя константы	Описание	Длина ключа
<code>ENCRYPT_DES</code>	Data Encryption Standard (DES)	56
<code>ENCRYPT_3DES_2KEY</code>	Modified Triple Data Encryption Standard (3DES); обрабатывает каждый блок трижды, используя 2 ключа	112
<code>ENCRYPT_3DES</code>	Triple Data Encryption Standard (3DES); обрабатывает каждый блок трижды	156
<code>ENCRYPT_AES128</code>	Advanced Encryption Standard	128
<code>ENCRYPT_AES192</code>	Advanced Encryption Standard	192
<code>ENCRYPT_AES256</code>	Advanced Encryption Standard	256
<code>ENCRYPT_RC4</code>	Потоковое шифрование (единственное)	

Типы сцепления DBMS_CRYPTO

Константа	Описание
CHAIN_CBC	Сцепление блоков шифротекста - Cipher Block Chaining
CHAIN_ECB	Электронная книга кодов - Electronic Code Book
CHAIN_CFB	Шифрование с обратной связью от шифротекста - Cipher Feedback
CHAIN_OFB	Шифрование с обратной связью по выходу - Output Feedback

При блочном шифровании данные обычно шифруются в блоках по восемь символов. Если длина входных данных не кратна восьми, добавляется недостающий символ или символы; этот процесс называется дополнением (padding). Простейший вариант – дополнение нулями, этот вариант включается константой PAD_ZERO, определенной в пакете DBMS_CRYPTO, но он не считается достаточно безопасным, поскольку потенциальный злоумышленник может угадать его. Более безопасное дополнение основано на стандарте Public-Key Cryptography Standard # 5 (PKCS#5, криптографический стандарт с общим ключом), задаваемый в пакете DBMS_CRYPTO константой PKCS5, как это показано в строке 8 листинга 1. Если длина данных уже кратна размеру блоков, то дополнения не потребуется, и это можно указать с помощью константы PAD_NONE. Чтобы выбрать метод дополнения, необходимо указать соответствующую константу из табл. 3 в значении параметра TYP (l_mod).

Т а б л и ц а 3

Типы дополнения DBMS_CRYPTO

Константа	Описание
PAD_PKCS5	Дополнение средствами криптографической системы с общим ключом (Public-Key Cryptography Standard # 5)
PAD_ZERO	Дополнение нулями
PAD_NONE	Отсутствие дополнения. Используется при уверенности в том, что длина уже кратна размеру шифруемого блока (кратно 8)

Все эти три параметра – алгоритм с длиной ключа, методы сцепления и дополнения – объединяются в один параметр TYP (l_mod), передаваемый во встроенную функцию шифрования ENCRYPT пакета DBMS_CRYPTO. Функции ENCRYPT требуется, чтобы незашифрованные данные имели тип RAW. Это преобразование делается в строке 11 листинга 1.

В пакете DBMS_CRYPTO есть функция дешифрования DECRYPT. Эта функция принимает для дешифрования исходные зашифрованные данные; ключ, использованный во время шифрования; а также объединенный параметр: алгоритм, длина ключа и схемы сцепления и дополнения. Вместе с данными, которые нужно дешифровать, необходимо передавать те же самые ключ и модификаторы, использованные во время шифрования.

Наиболее важным и сложным аспектом создания инфраструктуры шифрования является создание надежной и безопасной системы управления ключами, а не применение самих программных интерфейсов приложений.

Часто приходится использовать комбинацию ключей:

- один ключ для каждой строки БД; - мастер-ключ для всей БД.

В некоторых случаях требуется сравнение зашифрованных данных с входящими данными. Например, многие CRM-приложения (Customer Relationship Management - управление взаимоотношениями с клиентами) используют для уникальной идентификации клиентов такие атрибуты, как номера кредитных карт, номера паспортов, номера полисов и другие. Медицинским приложениям может потребоваться просмотр истории болезни для предложения плана лечения. Страховым компаниям может понадобиться просмотр диагнозов пациента для подтверждения справедливости жалоб. Все эти данные хранятся в зашифрованном виде, поэтому простое сравнение входящих данных с сохраненными данными невозможно.

Существуют два способа обработки подобных ситуаций:

1. Зашифровать поступившие данные и сравнить их с сохраненными зашифрованными данными. Этот метод реализуется только в случае, если известен ключ шифрования для подхода «один ключ для базы данных» (таблицы или схемы). Если был использован подход «новый ключ для каждой строки», то необходимо знать, какой ключ следует применить для шифрования значения в каждой конкретной строке.

2. Расшифровать зашифрованные данные в каждой строке и сравнить их с соответствующими открытыми данными. Для тех, кто использует отдельный ключ для каждой строки, этот способ является единственно возможным. Но каждая операция дешифрования потребляет несколько драгоценных циклов ЦПУ и может повлиять на общую производительность базы данных.

Как спроектировать систему так, чтобы сравнение с зашифрованными столбцами было наиболее эффективным? Предлагается следующее решение: выполнять сравнение не с зашифрованным значением, а с хеш-значением. Создание хеш-значения занимает значительно меньше времени, чем шифрование, и потребляет меньше циклов ЦПУ. Поскольку в результате хеширования некоторых данных всегда будет получено одно и то же хеш-значение, можно сохранить хеш-значение для конфиденциальных данных, создать хеш-значение для проверяемых на совпадение данных и сравнить его с сохраненным хеш-значением.

Предлагается следующая структура системы. В программном продукте «АВЕРС: Управление лесным фондом» есть таблица DOG_ARENDА для регистрации договоров аренды лесных участков, в которой хранятся номера договоров, требующие шифрования. Вместо того чтобы сохранять в таблице DOG_ARENDА номер договора, создадим две дополнительные таблицы ENC_ARENDА и ENC_KEYS (таблицы и связи между ними представлены на рис. 1).

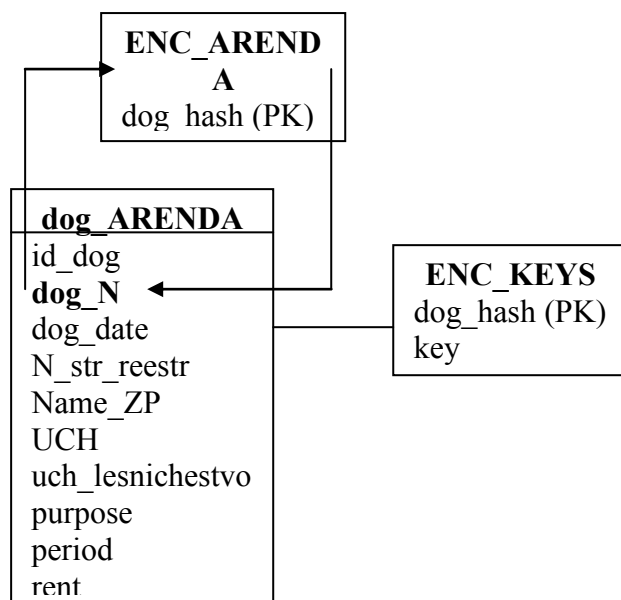


Рис. 1. Хранение зашифрованной информации о номерах договоров аренды лесных участков

Таблица dog_ARENDА содержит следующие атрибуты:

- id_dog – первичный ключ;
- dog_N – хеш-значение номера договора аренды, но не сам реальный номер;
- dog_date – дата государственной регистрации договора аренды лесного участка;
- N_str_reestr – кадастровый номер (номер учетной записи в государственном лесном реестре)
- Name_ZP – наименование землепользователя;
- UCH – лесной участок;
- uch_lesnichestvo – лесничество;
- purpose – цель аренды;
- period – срок аренды;
- rent – арендная плата.

Таблица ENC_ARENDА:

- dog_HASH – первичный ключ (хеш-значение договора),
- dog_ENC – зашифрованное значение номера договора аренды.

Таблица ENC_KEYS:

- dog_HASH – первичный ключ (хеш-значение договора),

KEY – ключ шифрования, используемый для шифрования данного номера договора аренды.

Незашифрованный номер договоров аренды нигде не хранится. Можно написать триггер, запускаемый перед вставкой (INSERT) или изменением (UPDATE) строки, который будет реализовывать такой код (листинг 2):

Л и с т и н г 2. Триггер шифрования TR_ARENDА

```
1 CREATE OR REPLACE TRIGGER tr_arena
2 before insert or update on
3 DOG_ARENDА
4 for each row
5 declare
6   l_mod VARCHAR2(255) := 'AES128';
7   l_hash varchar2(64);
8   l_enc raw(2000);
9   l_key raw(2000);
10 BEGIN
11   l_hash := get_hash_val(:NEW.dog_N);
12 BEGIN
13   select dog_enc INTO l_enc from ENC_ARENDА
14   where dog_hash=l_hash;
15 EXCEPTION
16 WHEN NO_DATA_FOUND
17 THEN
18   begin
19     l_key := dbms_crypto.randombytes (128/8);
20     l_enc := get_enc_val(
21       :NEW.dog_N,
22       l_key ,
23       l_mod);
24     INSERT INTO ENC_ARENDА values (l_hash,l_enc);
25     INSERT INTO ENC_KEYS values (l_hash,l_key);
26   end;
27 WHEN OTHERS
28 THEN
29   RAISE;
30 END;
31 :NEW.dog_N := l_hash;
32 END;
```

Триггер заменяет открытое значение хеш-значением, поэтому изменение приложения не требуется. Программы, сравнивающие номера договоров, будут искать совпадение с хеш-значением, а не с открытым или за-

шифрованными значениями. Алгоритм триггера шифрования TR_ARENDА показан на рис. 2.

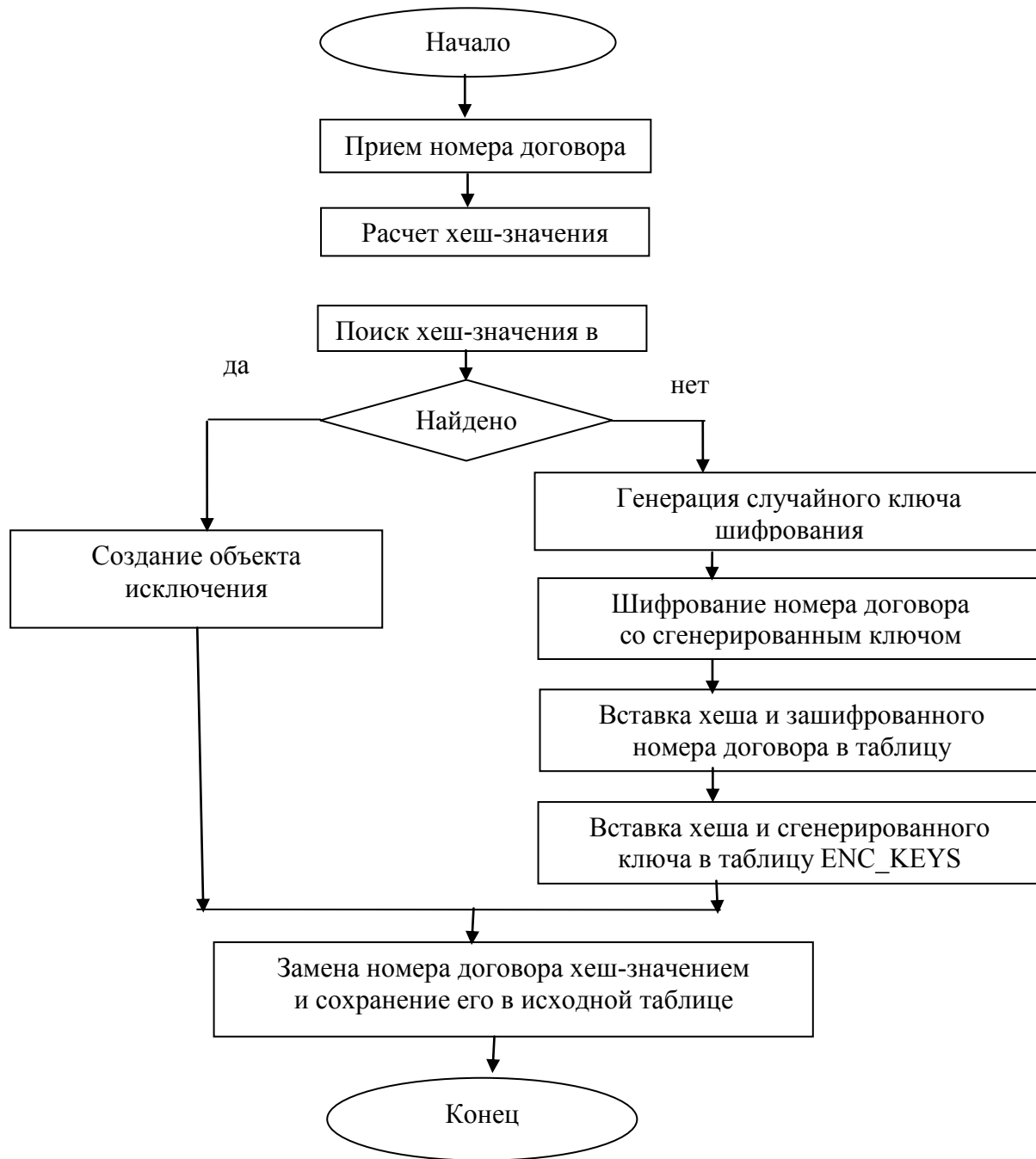


Рис. 2. Алгоритм триггера шифрования TR_ARENDА

В приведенном листинге 2 используется функция `get_hash_val` для выполнения хеширования по алгоритму SHA-1 (листинг 3):

Листинг 3. Функция хеширования

```
1 CREATE OR REPLACE FUNCTION get_hash_val (  
2 p_in_val IN VARCHAR2,  
3 p_algorithm IN VARCHAR2 := 'SH1')  
4 RETURN VARCHAR2  
5 IS  
6 l_hash_val RAW (2000);  
7 l_in RAW (2000);  
8 l_hash_algo PLS_INTEGER;  
9 l_ret VARCHAR2 (2000);  
10 BEGIN  
11 l_hash_algo:=  
12 CASE p_algorithm  
13 WHEN 'SH1' THEN DBMS_CRYPTO.HASH_SH1  
14 WHEN 'MD4' THEN DBMS_CRYPTO.HASH_MD4  
15 WHEN 'MD5' THEN DBMS_CRYPTO.HASH_MD5  
16 END;  
17 l_in:=UTL_I18N.STRING_TO_RAW(p_in_val,'AL32UTF8');  
18 l_hash_val:=DBMS_CRYPTO.HASH(  
19 SRC => l_in,  
20 TYP => l_hash_algo);  
21 l_ret := RAWTONHEX(l_hash_val);  
22 RETURN l_ret;  
23 END;
```

В целях стандартизации во всех приложениях применяют сцепление блоков шифротекста CBC и дополнение PKCS #5. Используя эти значения, можно создать общую функцию шифрования `GET_ENC_VAL`, показанную на листинге 4, которая имеет два обязательных входных параметра – исходные незашифрованные данные и ключ – и возвращает зашифрованные данные. Следует отметить, что эта функция может (необязательно) принимать на вход алгоритм шифрования и вектор инициализации.

Л и с т и н г 4. Функция шифрования

```
1 CREATE OR REPLACE FUNCTION get_enc_val (  
2 p_in_val IN VARCHAR2,  
3 p_key IN RAW,  
4 p_algorithm IN VARCHAR2 := 'AES128',  
5 p_iv IN VARCHAR2 := NULL)  
6 RETURN VARCHAR2
```

```

7 IS
8 l_enc_val RAW (2000);
9 l_enc_algo PLS_INTEGER;
10 l_in RAW (2000);
11 l_iv RAW (2000);
12 l_ret VARCHAR2 (2000);
13 BEGIN
14 l_enc_algo :=
15 CASE p_algorithm
16 WHEN 'DES' THEN dbms_crypto.ENCRYPT_DES
17 WHEN '3DES' THEN dbms_crypto.ENCRYPT_3DES
18 WHEN '3DES_2KEY' THEN dbms_crypto.ENCRYPT_3DES_2KEY
19 WHEN 'AES128' THEN dbms_crypto.ENCRYPT_AES128
20 WHEN 'AES192' THEN dbms_crypto.ENCRYPT_AES192
21 WHEN 'AES256' THEN dbms_crypto.ENCRYPT_AES256
22 WHEN 'RC4' THEN dbms_crypto.ENCRYPT_RC4
23 END;
24 l_in := UTL_I18N.STRING_TO_RAW (p_in_val, 'AL32UTF8');
25 l_iv := UTL_I18N.STRING_TO_RAW (p_iv, 'AL32UTF8');
26 l_enc_val := dbms_crypto.encrypt (
27 SRC => l_in,
28 KEY => p_key,
29 iv => l_iv,
30 TYP => l_enc_algo
31 + dbms_crypto.CHAIN_CBC
32 + dbms_crypto.PAD_PKCS5);
33 l_ret := RAWTOHEX (l_enc_val );
34 RETURN l_ret;
35 END;

```

Чтобы соединить три связанные таблицы (рис. 1) для получения расшифрованных данных, создадим представление VW_ARENDА, показанное на листинге 5. Данные расшифровываются с помощью функции GET_DEC_VAL (строка 6 листинга 5), приведенной в листинге 6. Эта функция возвращает данные типа VARCHAR2, поэтому они показываются как столбец VARCHAR2(2000); функция CAST в строке 5 листинга 5 преобразует их в данные типа VARCHAR2(30). Это представление как раз есть то, что предоставляется для доступа другим пользователям.

Л и с т и н г 5. Представление VW_ARENDА

```

1 create or replace view
2 vw_ARENDА
3 as
4 select

```



```

5 cast (
6 get_dec_val (e.dog_enc, k.key)
7 as varchar2(30)) as dog_N,
8 m.dog_date as dog_date,
9 m.N_str_reestr as N_str_reestr,
10 m.Name_ZP as Name_ZP,
11 m.UCH as UCH,
12 m.uch_lesnichestvo as uch_lesnichestvo,
13 m.purpose as purpose,
14 m.period as period,
15 m.rent as rent
16 from
17 dog_ARENDА m,
18 ENC_ARENDА e,
19 ENC_KEYS k
20 where
21 k.dog_HASH = e.dog_HASH and m.dog_N = e.dog_HASH;

```

Если злоумышленник получит доступ к исходной таблице DOG_ARENDА, то он не сможет воспользоваться зашифрованной информацией. Аналогично шифруются персональные данные арендаторов. Пользователи не могут видеть данные непосредственно в таблицах, а только через представление, поэтому информация защищена. Чтобы показывать только зашифрованные данные, защищая исходное содержимое, можно создавать различные представления базовой таблицы.

Л и с т и н г 6. Функция дешифрования

```

1 CREATE OR REPLACE FUNCTION get_dec_val (
2 p_in_val IN VARCHAR2,
3 p_key IN RAW,
4 p_algorithm IN VARCHAR2 := 'AES128',
5 p_iv IN VARCHAR2 := NULL)
6 RETURN VARCHAR2
7 IS
8 l_dec_val RAW (2000);
9 l_enc_algo PLS_INTEGER;
10 l_in RAW (2000);
11 l_iv RAW (2000);
12 l_ret VARCHAR2 (2000);
13 BEGIN
14 l_enc_algo :=
15 CASE p_algorithm
16 WHEN 'DES' THEN dbms_crypto.ENCRYPT_DES
17 WHEN '3DES' THEN dbms_crypto.ENCRYPT_3DES

```

```

18 WHEN '3DES_2KEY' THEN dbms_crypto.ENCRYPT_3DES_2KEY
19 WHEN 'AES128' THEN dbms_crypto.ENCRYPT_AES128
20 WHEN 'AES192' THEN dbms_crypto.ENCRYPT_AES192
21 WHEN 'AES256' THEN dbms_crypto.ENCRYPT_AES256
22 WHEN 'RC4' THEN dbms_crypto.ENCRYPT_RC4
23 END;
24 l_in := HEXTORAW (p_in_val);
25 l_iv := UTL_I18N.STRING_TO_RAW (p_iv, 'AL32UTF8');
26 l_dec_val := dbms_crypto.decrypt(
27 SRC    => l_in,
28 KEY    => p_key,
29 iv     => l_iv,
30 TYP    => l_enc_algo
31 + dbms_crypto.CHAIN_CBC
32 + dbms_crypto.PAD_PKCS5);
33 l_ret:= UTL_I18N.RAW_TO_CHAR(l_dec_val,'AL32UTF8');
34 RETURN l_ret;
35 END;

```

Следует отметить, что СУБД Oracle Database 10g Release 2 позволяет применять новую возможность прозрачного шифрования – Transparent Data Encryption (TDE), обеспечивающую прозрачное шифрование и дешифрование данных перед их сохранением в полях данных, используя главный ключ, который хранится в безопасном месте, называемом бумажником (wallet).

Библиографический список

1. Малыхина Г. Ф. Администрирование данных информационных систем: учеб. пособие / Г. Ф. Малыхина, Н. Г. Полетаева. – СПб.: Изд-во Политехн. ун-та, 2010. – 198 с.
2. Малыхина Г. Ф. Безопасность распределенных баз данных: учеб. пособие / Г. Ф. Малыхина, Н. Г. Полетаева. – СПб.: Изд-во Политехн. ун-та, 2012. – 195 с.
3. Нанда А. Oracle PL/SQL для администраторов баз данных: [пер. с англ.] / А. Нанда, С. Фейерштейн. – СПб.: Символ-Плюс, 2008. – 496 с.
4. Поляков А. М. Безопасность Oracle глазами аудитора: нападение и защита / А. М. Поляков. – М.: ДМК Пресс, 2010. – 336 с.

С.П. Хабаров, кандидат технических наук, доцент
Е.И. Шалаев, студент 3 курса
С.П. Васильев, студент 3 курса

РЕАЛИЗАЦИЯ МОДУЛЯ ЛОГИЧЕСКОГО ВЫВОДА В СОСТАВЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ КЛАССИФИКАЦИИ РАСТЕНИЙ

Решена задача включения в состав информационной системы классификации растений компонента экспертной системы продукционного типа. Обоснован подход к процессу формализации знаний и составу программных модулей, необходимых для данного класса систем.

Введение

Системами, основанными на знаниях, называют программные продукты, основными структурными элементами которых являются базы знаний (БЗ) и механизм логических выводов [1, 2]. К ним, в первую очередь относятся экспертные системы (ЭС), которые работают так, что имитируют образ действий человека-эксперта. При этом их работа существенно отличается от точных, хорошо аргументированных алгоритмов [3].

В рамках построения информационной системы классификации растений встала задача включения в её состав экспертной системы продукционного типа, которая бы реализовала как прямую, так и обратную цепочки вывода. Для этого потребовалось разработать ряд программных модулей (рис. 1):

- Модуль создания базы знаний и модуль загрузки базы в систему;
- Модуль работы с данными (добавления, редактирования и обработки данных), состоящий из экспертной и пользовательской части;
- Интерфейс связи ядра экспертной системы с диалоговыми формами.

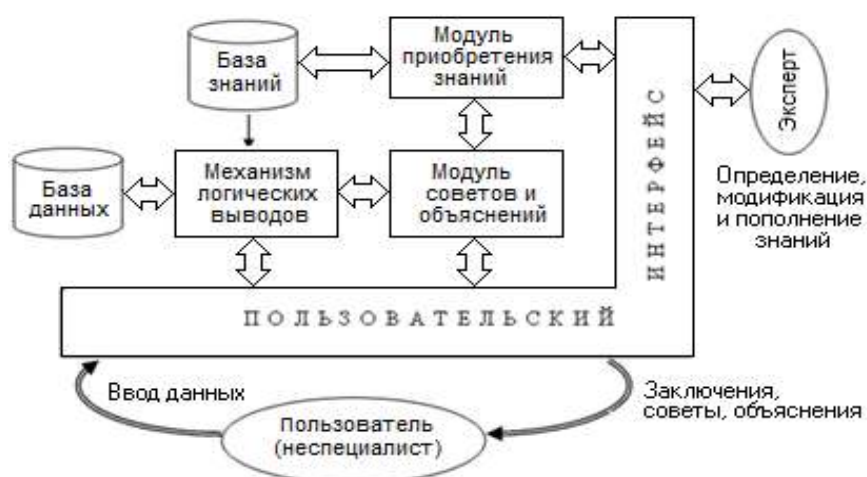


Рис. 1. Базовая структура экспертной системы.

Продукционная модель является наиболее распространенной в системах основанных на знаниях. Их отличает простота добавления и модификации

знаний, а также простота и точность механизма использования знаний ввиду их однородности и возможности использования единого синтаксиса для описания знаний. Основными элементами продукционных систем являются:

- База правил – это набор правил, используемый как база знаний.
- Рабочая память – хранит предпосылки, касающиеся конкретных задач предметной области, и результаты выводов, полученных на их основании.
- Механизм логического вывода.

Решение поставленной задачи потребовало обосновать подход к процессу формализации знаний, а также обосновать и разработать необходимый состав программных модулей для систем данного класса, уделив особое внимание реализации механизма логического вывода.

Процесс приобретения и формализации знаний

Важным этапом создания БЗ является этап приобретения знаний. На этом этапе разнообразный набор фактов должен быть представлен в виде некоторой обобщенной структуры. Одной из возможных является структура, получившая название дерево решений. Это один из простейших способов представления фактов. Его применение ограничено, но оно может быть эффективно там, где знания представляются в виде правил.

Наблюдаемая со стороны удаленная консультация специалиста в области ботаники по классификации некоторого растения выглядит как задаваемая им определенная последовательность вопросов для получения сведений, необходимых для решения задачи. Один из вариантов такой консультации может быть представлен в графической форме (рис. 2).

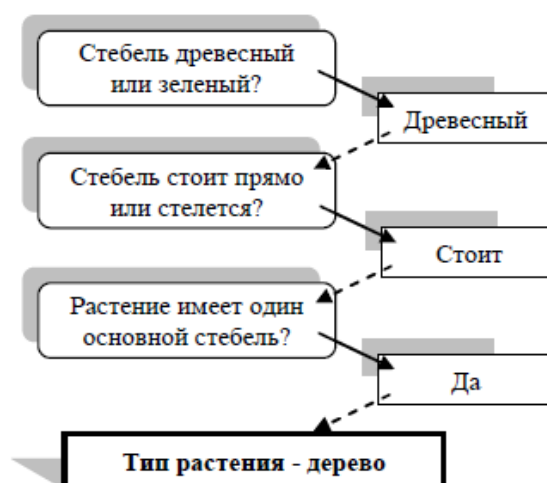


Рис.2. Вариант удаленной консультации эксперта.

Результат такой консультации: "Основываясь на Ваших ответах, можно предположить, что тип растения – это дерево". Однако, это только один из возможных вариантов экспертизы, проводимой экспертом. Аналогично могут быть представлены и все другие варианты консультаций. Что же позволяет человеку провести такую консультацию, определить как по-

следовательность, так и содержание вопросов в зависимости от ответов на них? Естественный ответ – это знания человека в конкретной предметной области, в которой он является специалистом – экспертом. Базируясь на его знаниях, диаграмму всех возможных исходов такой консультации можно представить в виде структуры дерева решений (рис. 3).

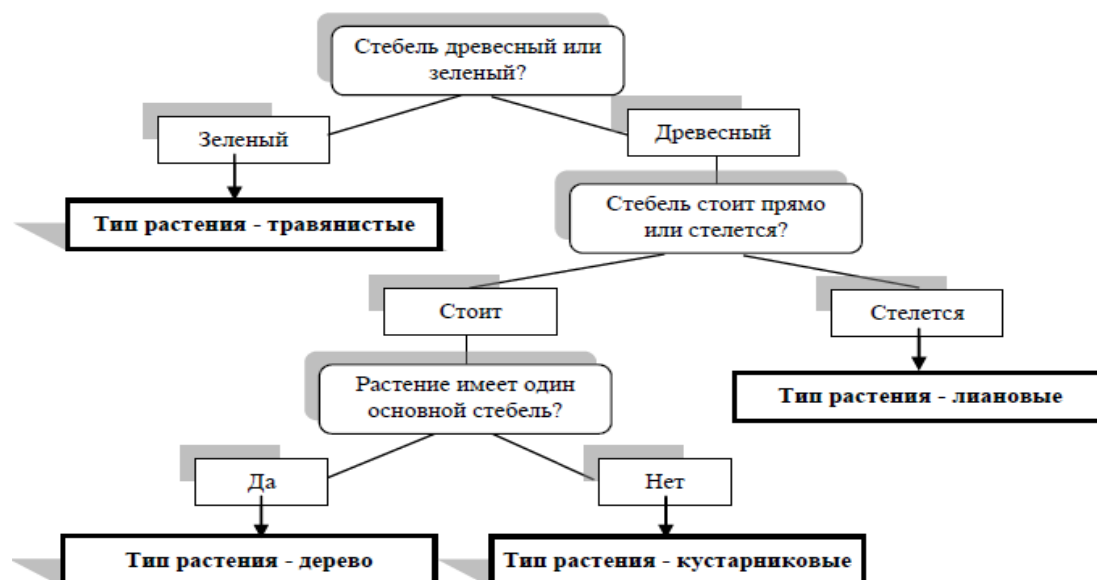


Рис. 3. Дерево решения для определения типа растения.

Но если консультация должна быть более глубокой и определять, например, класс растения, то в этом случае для вершины "Тип растения – дерево" должно быть построено свое дерево решений (рис. 4), которое будет поддеревом исходного дерева решений.

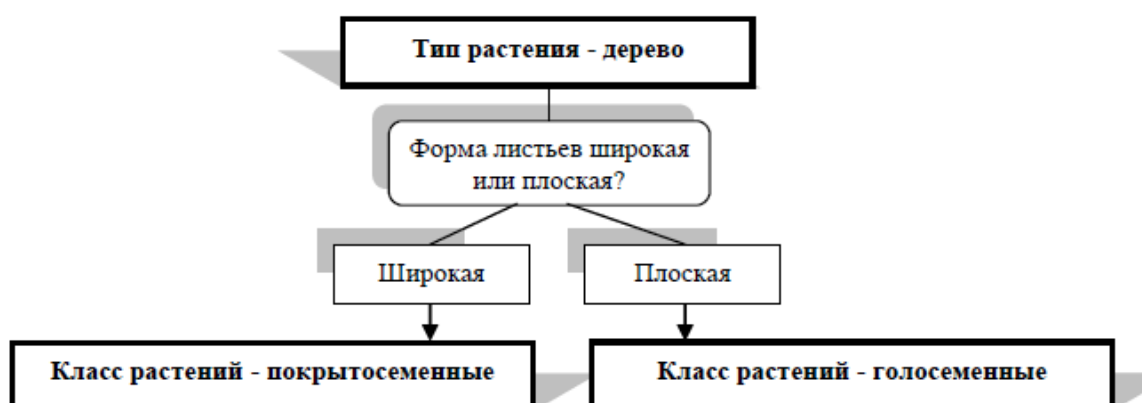


Рис. 4. Поддерево решения для определения класса растения.

Существует ряд причин, по которым полное дерево решений разбивается на отдельные секции. Во-первых, полное дерево решений быстро становится длинным и труднообозримым. Во-вторых, деление на отдельные секции существенно упрощает запоминание цели, которая преследуется в процессе извлечения знаний.

Когда какое-либо поддерево решений описано, то его заключительная часть может быть скопирована в корень вновь создаваемой ветви дерева решений, и для нее на основе знаний, получаемых от эксперта, может быть построено свое поддерево решений (рис. 5).



Рис. 5. Поддерево решения для определения семейства растения.

Процесс формализации должен продолжаться до тех пор, пока не будут детализированы все вершины поддерева решений. Таким образом, при разработке модели базы знаний продукционного типа необходимо:

- Общую задачу разбить на ряд подзадач.
- Для каждой подзадачи разработать свое дерево решений, что существенно упростит процесс создания и отладки базы знаний.
- Каждое дерево решений представить в виде совокупности правил.

Формирования правил осуществляется путем перевода дерева решений из модели вопросов-ответов в модель утверждения-факты. Для примера преобразование модели вопросов-ответов (рис. 3) в модель утверждения-факты будет иметь вид, аналогичный тому, что приведен на рис. 6.



Рис. 6. Перевод модели вопросов-ответов в утверждения-факты.

Аналогичные преобразования следует выполнить для всех поддеревьев. При этом все утверждения представляются предложениями, которые состоят из <атрибута, <предиката> и <значения>. В этой трактовке:

- Атрибут – это ключевое слово или фраза, описывающая некоторое качество, о котором мы пытаемся найти информацию.
- Значение – это описание, назначенное атрибуту.
- Предикат – указывает отношение между атрибутом и его значением.

Более строго эти понятия определены в логике предикатов. Здесь полагаем, что между атрибутом и его значением существует только одно отношение – отношение включения или совпадения. Эти отношения обычно описываются такими глаголами, как являться, принадлежать, быть. Набор правил на основе дерева решений формируется исходя из того, что:

- Каждая ветвь дерева решений от ее начала до конца образует правило;
- Совокупность предложений, которые расположены в модели утверждения-факты до стрелки, образуют условную часть правила, а после нее – заключение правила;
- Ветвь, в которой нет заключения, не может быть представлена в виде правила;
- Каждое предложение условной части правила представляет собой конъюнктивный член и соединяется логическим “И”.

В качестве примера можно привести одно из правил, которое получено из рис. 6 на основе приведенной выше методики:

ЕСЛИ стемель	– древесный
И положение	– прямое
И основной стемель	– один
ТО тип растения	– дерево

Такую модель продукционной базы знаний предлагается использовать в информационной системе классификации растений. Но для реализации этого подхода необходимо в составе системы иметь ряд программных модулей по созданию, ведению и редактированию базы знаний, механизму логического вывода на этой БЗ и внешнему интерфейсу к этой системе.

Подход к разработке оболочки продукционной ЭС

Специфика решаемой задачи по классификации растений во многом предопределила разработку собственной оболочки ЭС. Для ее реализации в проекте был использован объектно-ориентированный язык С# и технология XML. Создание набора программных модулей для собственной оболочки продукционных систем позволяет:

- Легко модифицировать ее и реализовывать различные варианты приложений: локальные, сетевые, мобильные, веб-приложение.
- Иметь удобный для создателя интерфейс, легко модифицируемый для конкретных задач под конкретного пользователя.

Кроме того, язык C# является превосходным инструментом, с помощью которого и начинающий пользователь, и программист профессионал могут создавать очень эффективные программы самых различных классов. К числу важных свойств, которые реализованы корпорацией Microsoft в языке программирования C#, можно отнести: компонентно-ориентированный подход к программированию, свойства как средство инкапсуляции данных, обработка событий, унифицированная система типизации, атрибуты и отражение, декларативные механизмы обеспечения безопасности выполнения кода, межъязыковое взаимодействие, достаточно полная и непротиворечивая система типов и т.д.

Для хранения и работы с данными, которые представляют собой базу знаний, необходимо сохранять их в структурированном виде. Для этого можно было бы использовать SQL сервер или что-то подобное. Однако, при желании переноса проекта на другие платформы, использование этих технологий могут вызвать дополнительные трудности. Поэтому для работы со структурированными данными был использован XML. Эта технология является кроссплатформенной, свободной от лицензирования и хорошо поддерживаемой.

Модуль создания базы знаний формирует шаблонный XML файл, который в последующем заполняется необходимыми данными. Для обработки XML файлов достаточно сложной структуры, когда значимой является информация элементов различного уровня вложенности, была использована библиотека LINQ to XML, которая расширяет возможности синтаксиса языка C# и предлагает стандартные, легко запоминающиеся шаблоны для выполнения запросов и обновления данных.

Структура XML файла состоит из трех блоков: объекты и их свойства (Obj), условные части правил (Rules), заключения правил (Result). Блоки Rules и Result имеют связь один к одному по идентификатору (рис. 7).


```

<Obj>
  <Item id="1" Object="Тип растения">
    <Properties Properties="Травянистые" />
    <Properties Properties="Дерево" />
    <Properties Properties="Кустарниковые" />
  </Item>
  <Item id="2" Object="Стебель">
    <Properties Properties="Зеленый" />
    <Properties Properties="Древесный" />
  </Item>
  <Item id="3" Object="Положение стебля">
    <Properties Properties="Прямое" />
    <Properties Properties="Стелется" />
  </Item>
  <Item id="4" Object="Основной стебель">
    <Properties Properties="Один" />
    <Properties Properties="Несколько" />
  </Item>
</Obj>

<Rules>
  <Item id="1">
    <Metod Object="Стебель" Properties="Зеленый" />
  </Item>
  <Item id="2">
    <Metod Object="Стебель" Properties="Древесный" />
    <Metod Object="Положение стебля" Properties="Стелется" />
  </Item>
  <Item id="3">
    <Metod Object="Стебель" Properties="Древесный" />
    <Metod Object="Положение стебля" Properties="Прямое" />
    <Metod Object="Основной стебель" Properties="Один" />
  </Item>
  <Item id="4">
    <Metod Object="Стебель" Properties="Древесный" />
    <Metod Object="Положение стебля" Properties="Прямое" />
    <Metod Object="Основной стебель" Properties="Несколько" />
  </Item>
</Rules>
<Results>
  <Item id="1" Object="Тип растения" Properties="Травянистые" />
  <Item id="2" Object="Тип растения" Properties="Травянистые" />
  <Item id="3" Object="Тип растения" Properties="Дерево" />
  <Item id="4" Object="Тип растения" Properties="Кустарниковые" />
</Results>
</BDName>

```

Рис. 7. Пример структуры XML файла и его заполнения.

Модуль загрузки текущей базы знаний считывает весь XML документ и загружает данные в структурированном виде в оперативную память

компьютера. Если XML файл имеет не корректную форму, то программа выведет ошибку о некорректной структуре файла.

Оболочка ЭС предусматривает работу разных категорий пользователей. При формировании БЗ доступно два режима: как формальное определение основных объектов предметной области с набором их свойств (эксперт-клерк), так и формирование на их основе набора правил (эксперт-профи), составляющих основу БЗ (рис. 8).

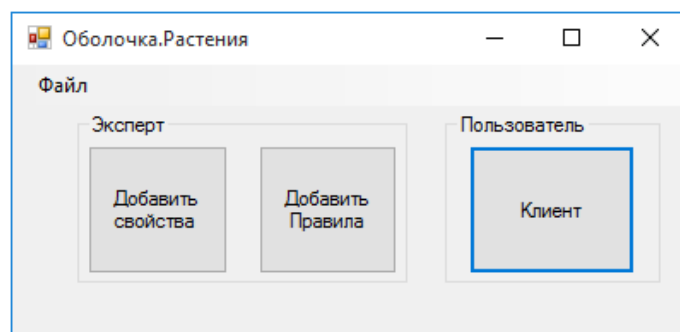


Рис. 8. Пример работы с модулем интерфейса.

Внешний интерфейс реализуется модулем работы с данными, который поддерживает как интерфейс-эксперта, так и интерфейс-пользователя. При этом интерфейс-эксперта реализует процедуры создания и модификации БЗ: описание объектов и их свойств, формированию правил (рис. 9 и 10).

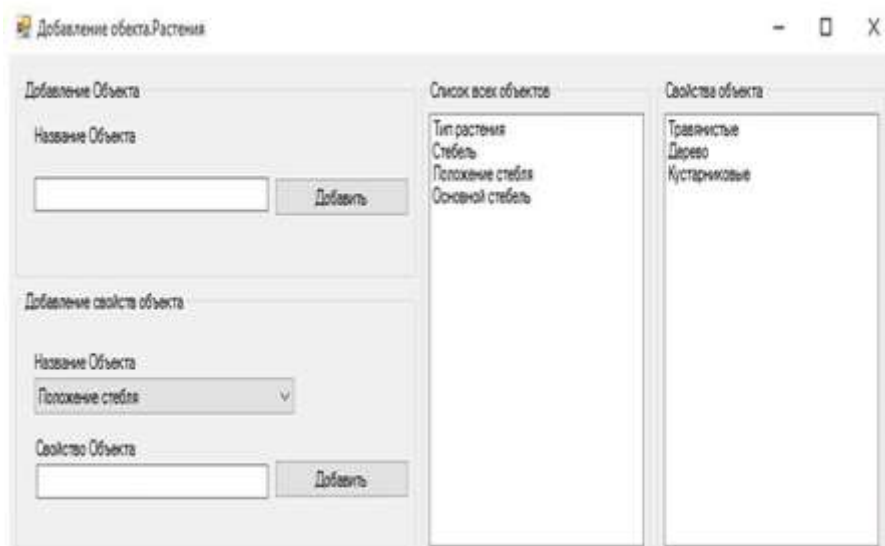


Рис. 9. Интерфейс эксперта – определение объектов и свойств

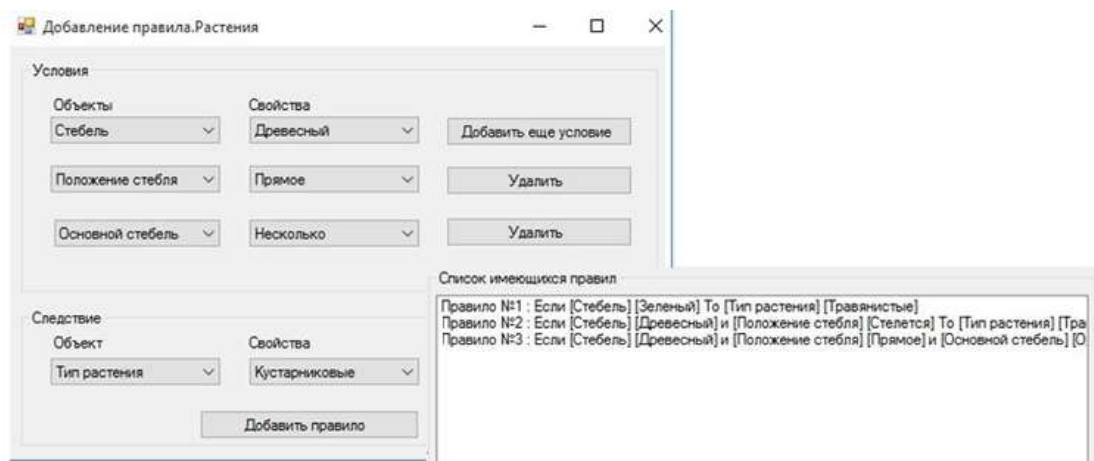


Рис. 10 Интерфейс эксперта – формирование набора правил.

Интерфейс пользователя предусматривает вывод результатов, исходя из правил, созданных экспертом. Допускает возможность логического вывода на основе как прямой, так и обратной цепочек рассуждения (рис. 11).

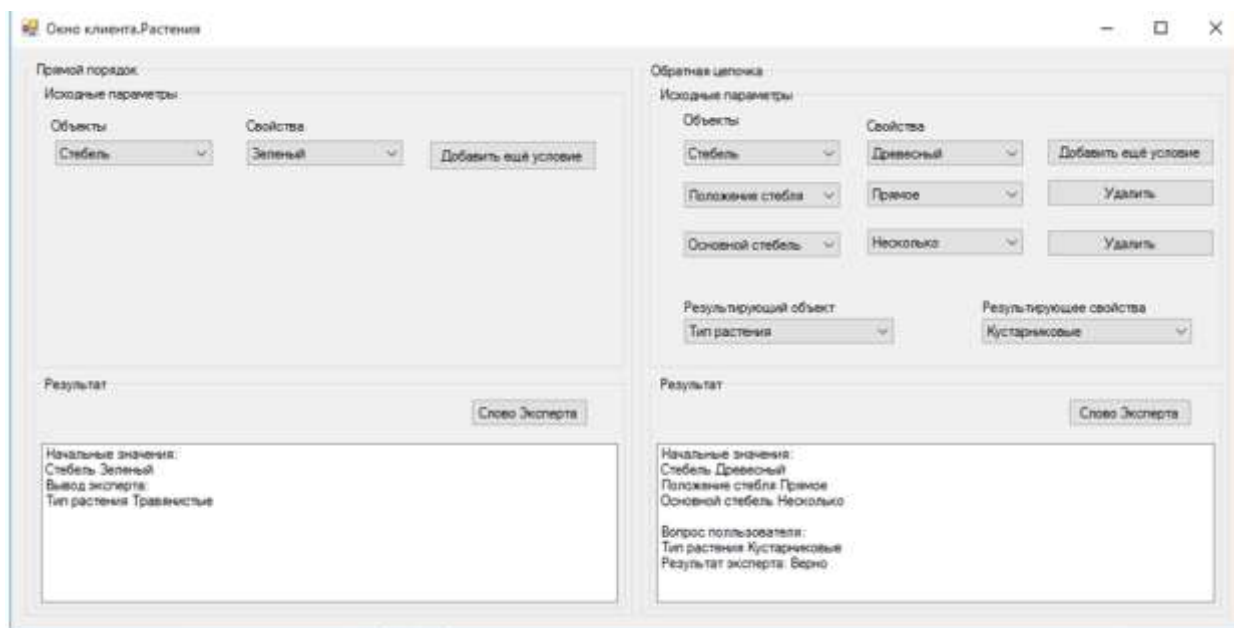


Рис. 11. Пример работы с интерфейсом пользователя.

После загрузки приложения считывается XML файл с правилами и строится программная модель экспертной системы: производится анализ файла на корректность, создаются списки переменных и правил, переменные инициализируются значениями из файла правил. Далее, в зависимости от режима работы системы, пользователю предоставляется следующие возможности:

- Просматривать правила и переменные, таблицы возможных решений;

- Производить консультацию с системой;
- Просматривать справки (информации о системе);
- Редактировать параметры системы (путь к файлу с правилами, порог активации правил и т.д.).
- Редактирование правил системы

Что касается модуля логического вывода, то он реализует стандартные для продукционных систем алгоритмы, которые сопоставляют образцы из условной части правила с фактами, хранимыми в рабочей памяти. Если все образцы имеются в рабочей памяти, то условная часть считается истинной, в противном случае – ложной. В процессе работы механизма логического вывода выполняется работа:

- По многократному просмотру содержимого базы правил;
- Последовательному применению правил на основе текущего состояния рабочей памяти;
- Дополнению данных, помещаемых в рабочую память.

На уровне управления последовательностью применения правил обычно выделяют две основные стратегии — применять правила в прямом или обратном порядке. Прямой порядок означает, что цепь рассуждений строится, отталкиваясь от данных (условий, о которых известно, что они удовлетворяются), к гипотезам (состоянию проблемы, вытекающему из этих условий). При обратном порядке рассуждения строятся, отталкиваясь от заданной цели (гипотезы, представляющие целевое состояние системы) к условиям, при которых возможно достижение этой цели.

Модуль логического вывода в свой состав включает механизм, который способен не только давать заключения, но и выводить всю цепочку вывода, приводящую к этим заключениям. Без этого пользователь может не понять заключение, которое он получил от экспертной системы.

Заключение

В рамках проекта по созданию информационной системы классификации лесных растений предлагается иметь в ее составе компонент экспертной системы продукционного типа. С этой целью предложен подход к процессу выявления и формализации экспертных знаний, а также обоснован состав программных модулей, необходимых для такого класса систем.

Специфика решаемой задачи предопределила разработку собственной оболочки ЭС. Для ее реализации в проекте был использован объектно-ориентированный язык C# и технология XML. Разработанные программные модули позволяют легко выполнять модификацию проекта, реализуя при этом различные варианты приложений, а также позволяют иметь удобный для разработчика интерфейс, который легко настраивается для конкретных задач под конкретного пользователя.

Библиографический список

1. Хабаров С. П. Интеллектуальные информационные системы. PROLOG - язык разработки интеллектуальных и экспертных систем. - СПб.: ЛТА, 2013. - 140 с.
2. Интеллектуальные системы и технологии Б. Я. Советов, В. В. Цехановский, В. Д. Чертовской. - М.: Издательский центр "Академия", 2013. - 320 с.
3. Хабаров С.П. Интеллектуальные информационные системы. [Электронный ресурс]. – Режим доступа: открытый ресурс; постоянный адрес в Интернет: – http://www.habarov.spb.ru/new_es/index.htm – Загл. с экрана.
4. Хабаров С. П., Жук Ю. А. Сетевые технологии взаимодействия Ubuntu и Windows платформ // С. П. Хабаров. — СПб.: Наука и техника, 2013.— 369 с.
- 5.

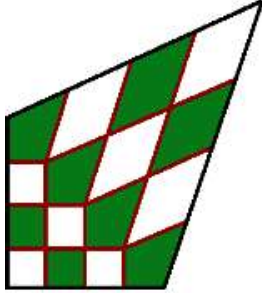
М.Л. Шилкина, кандидат технических наук, доцент

ПРИМЕНЕНИЕ РАЗЛИЧНЫХ ВИДОВ ПРЕОБРАЗОВАНИЙ КООРДИНАТ ДЛЯ КОРРЕКЦИИ КАРТОГРАФИЧЕСКИХ ИЗОБРАЖЕНИЙ

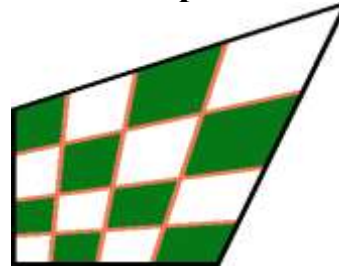
При решении задачи коррекции изображений, особенно связанных с картографией, возникают проблемы преобразований координат для четырёхугольных областей, переводящих их в четырёхугольные же области, но с другими координатами. Причем в общем случае прямоугольник не обязан переходить в прямоугольник или параллелограмм в параллелограмм. Подобные задачи возникают, если, например, картографическая бумага намочена и растянулась, или фотоаппарат при фотографировании карты стоял неровно, или бумага при сканировании замялась, во всех этих случаях необходимо провести коррекцию картографических изображений, чтобы привязка координат карты стала правильной. Другими словами, необходимо найти и применить к точкам четырёхугольной области изображения преобразование координат, переводящее эту область в, как правило, прямоугольную область с заданными координатами.

Существует три метода преобразований координат, сохраняющих прямые: аффинное, билинейное и перспективное преобразования (рис.1). Аффинные, кроме того, сохраняют после применения преобразований параллельные прямые параллельными.

Аффинное преобразование координат



Перспективное преобразование координат



Билинейное преобразование координат

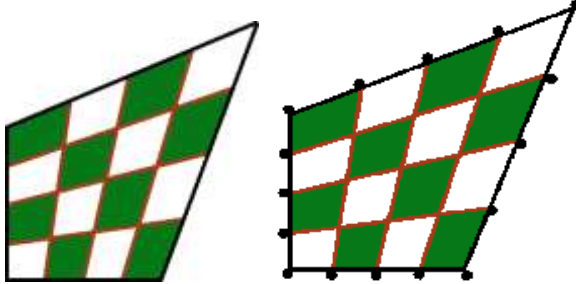


Рис.1. Преобразования координат, сохраняющие прямые

Любое аффинное преобразование координат является суперпозицией четырёх основных видов таких преобразований - параллельного переноса, сдвига, масштабирования и поворота. В матричном виде в однородных координатах аффинное преобразование:

$$\begin{pmatrix} x' & y' & 1 \end{pmatrix} = \begin{pmatrix} x & y & 1 \end{pmatrix} \cdot \begin{pmatrix} a & c & 0 \\ b & d & 0 \\ l & m & 1 \end{pmatrix}$$

При применении нескольких аффинных преобразований результирующая матрица преобразования равна произведению матриц всех последовательных преобразований. Очевидно, что это преобразование обратимо, матрица обратного преобразования равна обратной матрицы исходного.

Возвращаясь к рассматриваемой картографической задаче, проблема в ней заключается в том, что на карте нанесены координатные сетки, которые с некоторым приближением можно считать прямоугольными. После сканирования эта координатная сетка на изображении получается с неким искажением, сдвигом и т.д. Физические координаты некоторых точек на изображении карты известны - эти точки называют реперными, их координаты на искаженном изображении можно узнать, например, в графическом редакторе, с другой стороны, для них известны координаты, которые они должны будут иметь на исправленной карте с правильной координатной

сеткой. Реперные точки - линии пересечения прямоугольной координатной сетки.

Тогда возникает задача нахождения аффинного преобразования, переводящего один четырехугольник в другой. Для этого все преобразуемые четырехугольники разбиваются диагоналями на треугольники, для каждого из которых можно найти соответствующее условию задачи аффинное преобразование (рис.2). Т.о. пусть заданы точки соответствия в исходном и конечном треугольниках:

$$\begin{aligned} (x_0 \quad y_0) &\leftrightarrow (x'_0 \quad y'_0) \\ (x_1 \quad y_1) &\leftrightarrow (x'_1 \quad y'_1) \\ (x_2 \quad y_2) &\leftrightarrow (x'_2 \quad y'_2) \end{aligned}$$

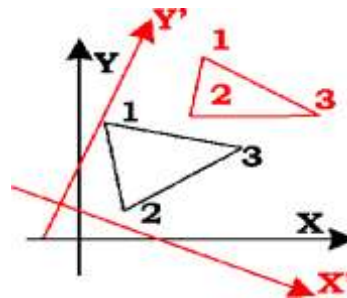


Рис.2. Задача нахождения аффинного преобразования координат по заданным парам точек.

Требуется найти матрицу преобразования координат, отображающую пары точек соответствия согласно заданному условию:

$$P = P' \cdot M, \quad M = ?$$

В матричном виде [1]:

$$\begin{pmatrix} x_0 & y_0 & 1 \\ x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \end{pmatrix} = \begin{pmatrix} x'_0 & y'_0 & 1 \\ x'_1 & y'_1 & 1 \\ x'_2 & y'_2 & 1 \end{pmatrix} \cdot \begin{pmatrix} a & c & 0 \\ b & d & 0 \\ l & m & 1 \end{pmatrix}$$

$$G = G' \cdot M$$

$$\begin{pmatrix} a & c & 0 \\ b & d & 0 \\ l & m & 1 \end{pmatrix} = \frac{1}{\det G'} \cdot \begin{pmatrix} y'_1 - y'_2 & y'_2 - y'_0 & y'_0 - y'_1 \\ x'_2 - x'_1 & x'_0 - x'_2 & x'_1 - x'_0 \\ x'_1 y'_2 - x'_2 y'_1 & x'_2 y'_0 - x'_0 y'_2 & x'_2 y'_1 - x'_1 y'_2 \end{pmatrix} \cdot \begin{pmatrix} x_0 & y_0 & 1 \\ x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \end{pmatrix}$$

$$\text{здесь: } \det G' = x'_0 \cdot (y'_1 - y'_2) - y'_0 \cdot (x'_1 - x'_2) + (x'_1 y'_2 - x'_2 y'_1)$$

Теперь, зная матрицу искомого преобразования координат, можно вычислить исправленные координаты для изображения карты, на котором координатная сетка будет отображаться корректно. Недостаток аффинного преобразования можно видеть на рис.1 - на границе треугольных областей хотя и сохраняется непрерывность изображения, но тем не менее возникают артефакты, связанные с особенностями аффинных преобразований, сохраняющими после преобразования параллельные прямые параллельными. Поэтому в центре четырехугольной области после преобразования образуются зоны неестественного растяжения-сжатия (рис.3), т.е. хотелось бы

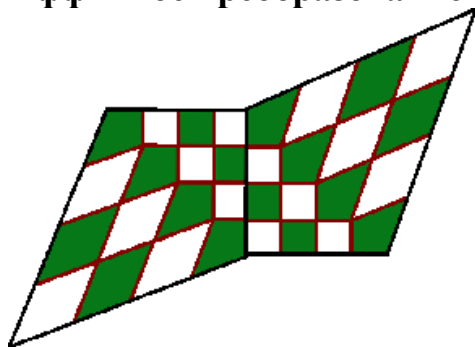
иметь более гладкую коррекцию изображения, но с сохранением непрерывности изображения не границах зон соответствия и правил отображения реперных точек координатной сетки.



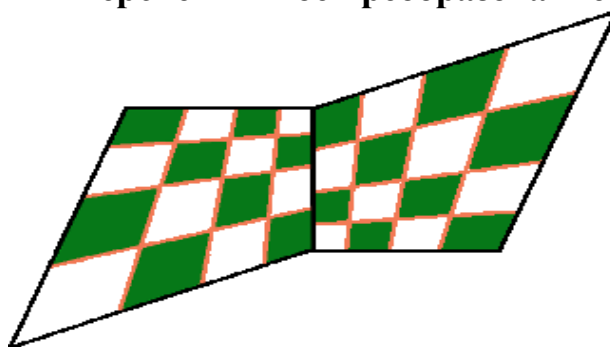
Рис.3. Результат применения аффинного преобразования координат

Альтернативным подходом является использование билинейного преобразования координат, обладающего сглаживающим эффектом по отношению к артефакту наличия неестественных резких переходов на границах треугольников разбиения от областей растяжения изображения к участкам сжатия (как в аффинных преобразованиях рис.1). При этом в билинейном преобразовании сохраняется непрерывность изображения на стыке четырехугольников соответствия (рис.4).

Аффинное преобразование



Перспективное преобразование



Билинейное преобразование

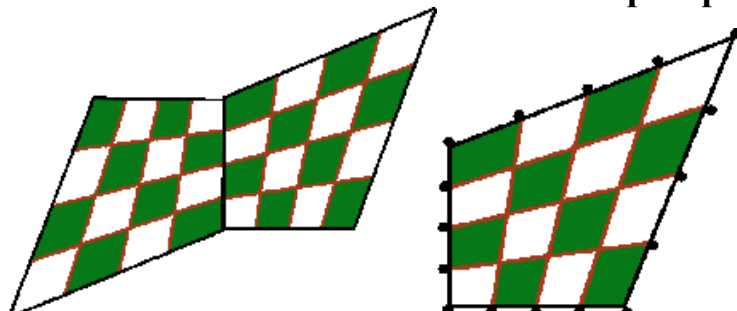


Рис.4. Сопряжение областей соответствия для различных преобразований координат

Суть билинейного преобразования состоит в равномерном разбиении противоположных сторон четырехугольников соответствия на одинаковое число частей. Затем мысленно соединив точки разбиения на противоположных сторонах, получаем сетку разбиения (рис.4). Частоту разбиения можно выбрать так, чтобы в одной ячейке сетки разбиения в обеих областях соответствия (на искаженном и на исправленном изображении) помещалось не более одного пикселя. Тогда используя для вычисления координат точек внутри области формулу линейной интерполяции можно изменяя параметры t и s , пробежать все точки четырехугольника исходного изображения и переписать их цвет в соответствующие им точки четырехугольника скорректированного изображения, у которых будет такое же значение параметров t и s (рис.5).

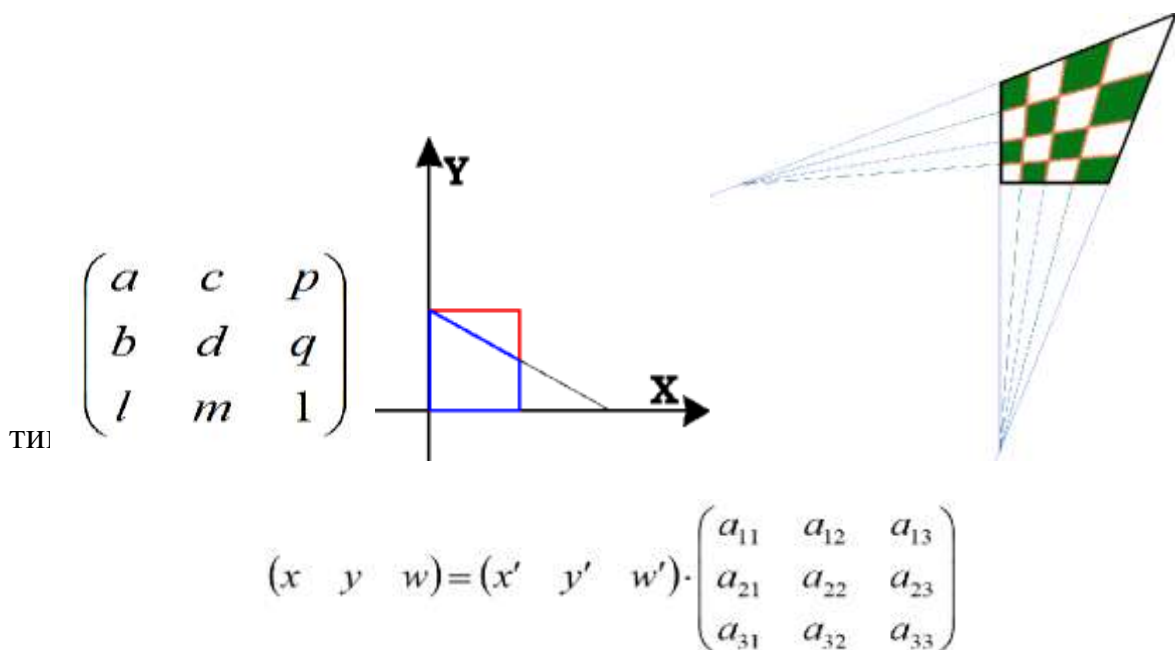


Рис.5. Результат коррекции картографического изображения при помощи билинейного преобразования координат

$$\begin{aligned}
X &= X_0 \cdot (1-t) + X_1 \cdot t, & t \in [0,1] \\
Y &= Y_0 \cdot (1-s) + Y_1 \cdot s, & s \in [0,1] \\
X' &= X'_0 \cdot (1-t) + X'_1 \cdot t, & t \in [0,1] \\
Y' &= Y'_0 \cdot (1-s) + Y'_1 \cdot s, & s \in [0,1]
\end{aligned}$$

На рис.5 виден результат программной обработки снимка карты (имевшего как механические дефекты бумаги, так и неправильное позиционирование камеры при съёмке), на котором координатная сетка приведена к корректному регулярному виду.

Еще одним видом преобразований координат, сохраняющим прямые, является перспективное преобразование (рис.6). Оно является обобщением аффинного преобразования, в результате применения которого прямые, параллельные на исходном изображении, оказываются в общем случае пересекающимися в некоей точке, называемой точкой схода, на результирующем изображении. Изучением особенностей этого преобразования занимается проективная геометрия. В отличие от матрицы аффинного преобразования, правый столбец матрицы перспективного преобразования не единичный.



Здесь координаты w и w' уже не обязаны в общем случае обе равняться 1, как в аффинном преобразовании, так как правый столбец матрицы преобразования неединичный. Но одну из них, координату w' всегда можно положить равной 1, просто разделив на нее обе части равенства. Тогда обычные (необобщенные) декартовы координаты полученного в результате преобразований образа:

$$x_{\text{обыч}} = \frac{x}{w} = \frac{a_{11}x' + a_{21}y' + a_{31}}{a_{13}x' + a_{23}y' + 1}, \quad y_{\text{обыч}} = \frac{y}{w} = \frac{a_{12}x' + a_{22}y' + a_{32}}{a_{13}x' + a_{23}y' + 1}$$

Умножим обе части на знаменатель и выразим x и y [2] и получим формулу расчета координат:

$$\begin{aligned} x &= a_{11}x' + a_{21}y' + a_{31} - a_{13}x'x - a_{23}y'x \\ y &= a_{12}x' + a_{22}y' + a_{32} - a_{13}x'y - a_{23}y'y \end{aligned}$$

Как и в случае аффинных преобразований, стоит задача нахождения матрицы перспективного преобразования (a_{ij}) по известным координатам точек соответствия.

$$\begin{aligned} (x_0 \quad y_0) &\leftrightarrow (x'_0 \quad y'_0) \\ (x_1 \quad y_1) &\leftrightarrow (x'_1 \quad y'_1) \\ (x_2 \quad y_2) &\leftrightarrow (x'_2 \quad y'_2) \\ (x_3 \quad y_3) &\leftrightarrow (x'_3 \quad y'_3) \end{aligned}$$

$$P = P' \cdot M, \quad M = ?$$

Подставим координаты точек соответствия в формулу расчета координат и получим систему из 8 уравнений и 8 неизвестных a_{ij} :

$$\begin{pmatrix} x'_0 & y'_0 & 1 & 0 & 0 & 0 & -x'_0x_0 & -y'_0x_0 \\ x'_1 & y'_1 & 1 & 0 & 0 & 0 & -x'_1x_1 & -y'_1x_1 \\ x'_2 & y'_2 & 1 & 0 & 0 & 0 & -x'_2x_2 & -y'_2x_2 \\ x'_3 & y'_3 & 1 & 0 & 0 & 0 & -x'_3x_3 & -y'_3x_3 \\ 0 & 0 & 0 & x'_0 & y'_0 & 1 & -x'_0y_0 & -y'_0y_0 \\ 0 & 0 & 0 & x'_1 & y'_1 & 1 & -x'_1y_1 & -y'_1y_1 \\ 0 & 0 & 0 & x'_2 & y'_2 & 1 & -x'_2y_2 & -y'_2y_2 \\ 0 & 0 & 0 & x'_3 & y'_3 & 1 & -x'_3y_3 & -y'_3y_3 \end{pmatrix} \cdot \begin{pmatrix} a_{11} \\ a_{21} \\ a_{31} \\ a_{12} \\ a_{22} \\ a_{32} \\ a_{13} \\ a_{23} \end{pmatrix} = \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ y_0 \\ y_1 \\ y_2 \\ y_3 \end{pmatrix}$$

Алгоритмически задачу нахождения матрицы перспективного преобразования проще решать в два этапа (рис.7). Сначала находим матрицу перспективного преобразования, отображающую исходный четырехугольник соответствия в единичный квадрат, а затем матрицу, переводящую единичный квадрат в результирующий четырехугольник соответствия. Затем перемножив эти две матрицы, получим матрицу отображения исходного четырехугольника соответствия в результирующий.

При этом, очевидно, удобнее на втором этапе найти матрицу отображения результирующего четырехугольника в единичный квадрат, а затем эту матрицу обратить, так как нам требуется матрица обратного преобразования. Тогда на обоих этапах надо решать одну и ту же задачу, только с разными исходными данными вида:

$$\begin{aligned}(x_0 \ y_0) &\leftrightarrow (0 \ 0) \\(x_1 \ y_1) &\leftrightarrow (1 \ 0) \\(x_2 \ y_2) &\leftrightarrow (1 \ 1) \\(x_3 \ y_3) &\leftrightarrow (0 \ 1)\end{aligned}$$

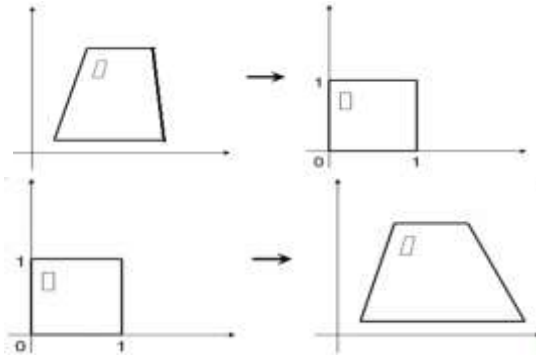


Рис.7. Два этапа нахождения матрицы перспективного преобразования.

Подставляем эти данные в систему уравнений:

$$\begin{aligned}a_{31} &= x_0 \\a_{11} + a_{31} - a_{13}x_1 &= x_1 \\a_{11} + a_{21} + a_{31} - a_{13}x_2 - a_{23}x_2 &= x_2 \\a_{21} + a_{31} - a_{23}x_3 &= x_3 \\a_{32} &= y_0 \\a_{12} + a_{32} - a_{13}y_1 &= y_1 \\a_{12} + a_{22} + a_{32} - a_{13}y_2 - a_{23}y_2 &= y_2 \\a_{22} + a_{32} - a_{23}y_3 &= y_3\end{aligned}$$

Введя обозначения:

$$\begin{aligned}\Delta x_1 &= x_1 - x_2 & \Delta x_2 &= x_3 - x_2 & \Delta x_3 &= x_0 - x_1 + x_2 - x_3 \\ \Delta y_1 &= y_1 - y_2 & \Delta y_2 &= y_3 - y_2 & \Delta y_3 &= y_0 - y_1 + y_2 - y_3\end{aligned}$$

Получим решение системы уравнений:

$$\begin{aligned}a_{13} &= \frac{\begin{vmatrix} \Delta x_3 & \Delta x_2 \\ \Delta y_3 & \Delta y_2 \end{vmatrix}}{\begin{vmatrix} \Delta x_1 & \Delta x_2 \\ \Delta y_1 & \Delta y_2 \end{vmatrix}} & a_{11} &= x_1 - x_0 + a_{13}x_1 \\ & & a_{21} &= x_3 - x_0 + a_{23}x_3 \\ & & a_{31} &= x_0 \\ a_{23} &= \frac{\begin{vmatrix} \Delta x_1 & \Delta x_3 \\ \Delta y_1 & \Delta y_3 \end{vmatrix}}{\begin{vmatrix} \Delta x_1 & \Delta x_2 \\ \Delta y_1 & \Delta y_2 \end{vmatrix}} & a_{12} &= y_1 - y_0 + a_{13}y_1 \\ & & a_{22} &= y_3 - y_0 + a_{23}y_3 \\ & & a_{32} &= y_0\end{aligned}$$

Зная матрицу перспективного преобразования, корректно отображающего точки соответствия (реперные), можно рассчитать новые координаты в результирующем изображении для каждого пикселя из исходного изображения, применив, при необходимости, интерполяцию цвета (рис.8).

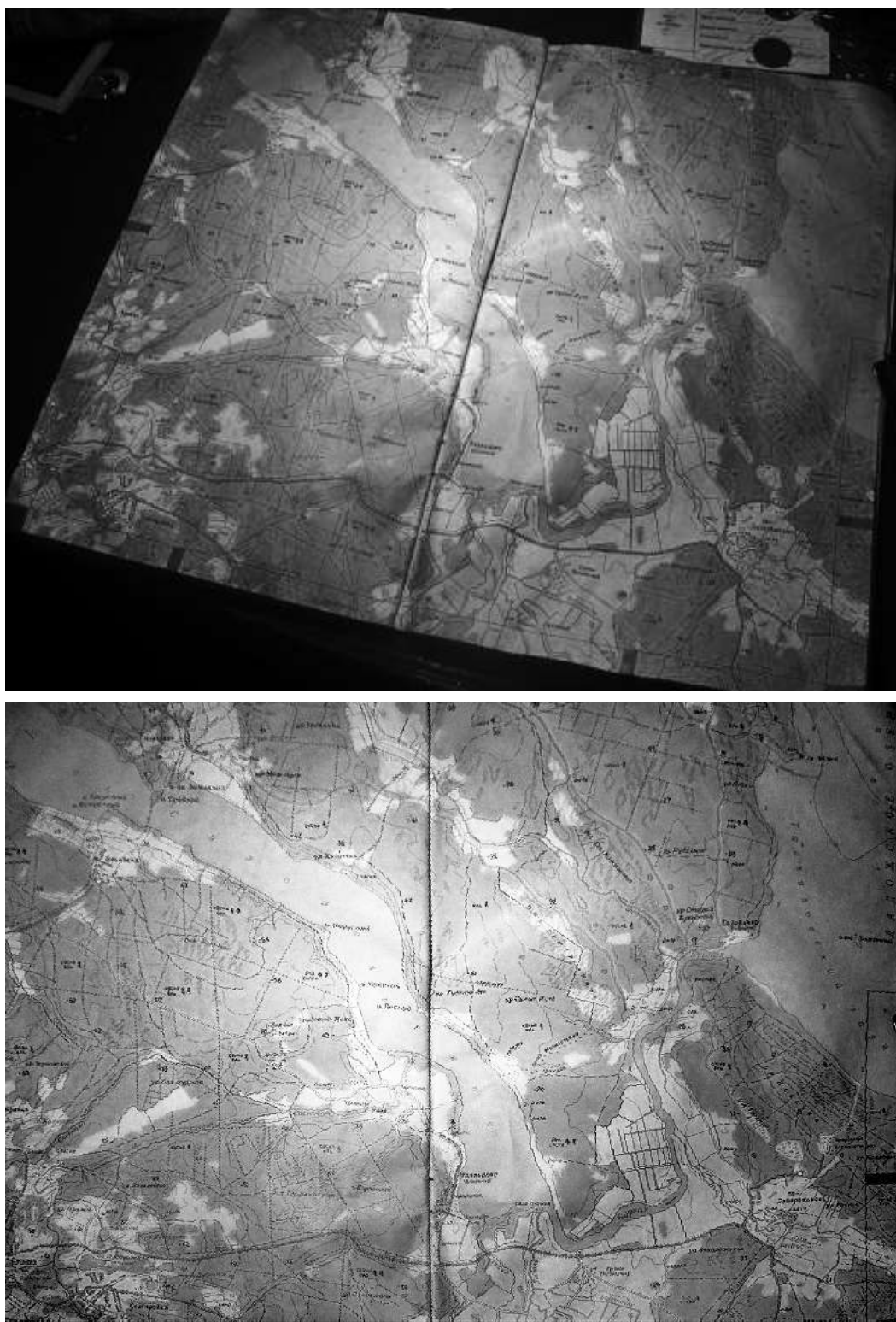


Рис.8. Применение перспективного преобразования к картографическому снимку

Преимуществом перспективного преобразования по сравнению с билинейным является возможность точной корректировки перспективных искажений снимка при неправильном позиционировании камеры, чего невозможно добиться применением билинейного или аффинного преобразования. С другой стороны, недостаток использования перспективного преобразования также очевиден - он не даёт возможности разбивать изображение на несколько зон соответствия, так как на границах зон после применения перспективного преобразования теряется непрерывность изображения (см.рис.3). Поэтому можно сделать вывод, что в случае механических деформаций бумаги сканируемого изображения для коррекции полученного снимка надо применять разбиение на зоны соответствия координатной сетки и применять к каждой зоне соответствующее ей билинейное преобразование, переводящее реперные точки в правильные физические координаты. Также можно заключить, что в некоторых случаях рационально будет применить сначала перспективное преобразование изображения для устранения дефектов съёмки, а затем к полученному изображению применить билинейное преобразование для ликвидации искажений, вызванных дефектами бумаги.

Библиографический список

1. David F. Rodgers, J. van Adams. "Mathematical Elements for Computer Graphics", 2nd ed., McGraw-Hill Publishing Company, 1990.
2. Alan Watt, Mark Watt. "Advanced Animation and Rendering Techniques. Theory and Practice", ACM Press, Addison-Wesley Longman Limited, 1992.

М.А. Шубина, кандидат технических наук, доцент

НОВЫЕ ПРОГРАММНЫЕ ПРОДУКТЫ ГИС-ТЕХНОЛОГИЙ

Расширение областей использования пространственно распределенной информации с необходимостью указания ее локализации (например, географических координат) привело к бурному развитию дистанционных методов ее получения (прежде всего, спутниковых) и обработки с использованием ГИС-технологий. В первую очередь, это касается исследования, планирования и мониторинга территорий, в том числе залесенных территорий.

Однако освоение этих технологий ограничено стоимостью приобретения лицензированных программных продуктов, хотя для обучения появились еще недостаточно используемые возможности.

В настоящее время ГИС-технологии реализуются с помощью достаточно большого количества программных продуктов: SAGAGIS (System for Automated Geoscientific Analyses, 1991-2015, разработчик кафедры географии, Геттинген), GRASS GIS (Geographic Resources Analysis Support System, 1995-2013, разработчик USA - CERL), QGIS (Quantum GIS, 2004-2014, разработчик Free Software Foundation, USA на основе ОС GNU), ARCGIS (2005-2015, разработчик ESRI, США), PHOTOMOD (1994-2014, разработчик МФТИ, Россия), ENVI (Environment for Visualizing Images, разработчик Exelis, бывшая ITT Visual Information Solutions, США), раньше всего появившиеся на рынке – ERDAS (ERDAS IMAGINE, 1978-2014, разработчики ERDAS, Leica Geosystem, ныне Intergraph, США), ERMapper (Earth Resource Mapping, разработчик ERDAS), MapInfo, Global Mapper (преобразована из Blue Marble Geographics (1995) в 2011, разработчик Global Mapper) и др.

Перечисленные продукты в течение 35 – 40 лет присутствуют на рынке и постоянно обновляются, выпуская все новые версии с учетом развития вычислительных систем, спутниковых систем, потребностей пользователей, хотя некоторые меняли компанию-разработчика.

Например, ERDAS IMAGINE 2013 года (это наиболее развитая из программ для обработки данных дистанционного зондирования (ДЗЗ)) предлагает новую технологию одновременного просмотра облака точек в 2D и 3D, нынешний владелец Intergraph объявил о выпуске ERDAS IMAGINE с динамичными инструментами, которые позволят пользователям более тщательно визуализировать, анализировать и управлять своими облачными точками [1].

Популярность той или иной ГИС-системы зависит от используемых операционных систем. В России чаще всего используются системы на основе ОС Windows, Unix, а в последнее время Linux. Последние версии все чаще конструируются, как открытые общедоступные системы, которыми могут пользоваться «широкие слои пользователей» Интернета.

По состоянию на 2014 год PHOTOMOD единственная поддерживаемая в актуальном состоянии ЦФС в России, не уступающая зарубежным аналогам. Семейство продуктов PHOTOMOD состоит из цифровой фотограмметрической системы и программ: PHOTOMOD GeoMosaic, PHOTOMOD UAS, PHOTOMOD Radar, PHOTOMOD Conveyor, PHOTOMOD Lite. Выходными продуктами ЦФС PHOTOMOD являются: пространственная фототриангуляция, цифровые модели рельефа и местности, двумерные и трехмерные вектора, ортофотопланы, 3D-модели, цифровые карты. Используются в таких прикладных областях как: картография, геодезия, археология, строительство, кадастр, лесном хозяйстве, изучении

небесных тел, биологических исследованиях. С 2014 года система переведена на новую платформу 64-битной архитектуры [2].

К числу основных компаний, обладающих правами на распространение популярных ГИС-программ на российском рынке, относятся «Совзонд», «Сканэкс» и «ДАТА+», которые знакомят пользователей и членов ГИС-ассоциации с новинками путем проведения вебинаров, школ, семинаров и конференций. Эти компании разрабатывают и свои продукты, и сотрудничают с крупными компаниями-потребителями РФ.

Одна из распространенных в России систем, с заслуживающими интереса пользователей новинками – ENVI, эксклюзивным дистрибьютором по распространению которой на территории России и стран СНГ является компания «Совзонд» [3].

Как известно, назначение ENVI – визуализация и обработка данных дистанционного зондирования; обработка и анализ мультиспектральных и гиперспектральных снимков; исправления геометрических и радиометрических искажений; интерактивное улучшение изображений; пространственная привязка изображения; ортотрансформирование; создание ЦМР на основе стереоизображений; трехмерная визуализация; интерактивное дешифрирование и классификация; анализ растительности с использованием индексов NDVI; анализа снимков в радиодиапазоне; векторизация растровых данных поддержка широкого диапазона растровых и векторных форматов; калибровка и атмосферная коррекция.

Область применения – обработка данных дистанционного зондирования, полученных со спутников QuickBird, IKONOS, WorldView-1/2/3, OrbView, GeoEye-1, Formosat-2, Cartosat-1, Pleiades-1, RESOURCESAT, EROS, IRS, Landsat, SPOT-6, RADARSAT, ENVISAT, MODIS, Terra (ASTER), EO1 (Hyperion) и др.; приложения для ГИС и данных цифровой картографии.

Открытая архитектура ENVI обеспечивает удобство обработки и предоставляет пользователю возможность индивидуальной настройки с помощью языка программирования IDL.

В отличие от других пакетов по обработке снимков программный комплекс ENVI написан на языке программирования IDL (Interactive Data Language), позволяющем создавать собственные алгоритмы обработки данных, а также предоставляет пользователю возможность индивидуальной настройки ENVI и модификацию пользовательского интерфейса.

К достоинствам системы относят широкий выбор входных и выходных форматов; отсутствие ограничений по размеру обрабатываемых изображений; автоматическая и ручная векторизация растровых данных, динамический доступ к данным ГИС и СУБД; подготовка карт, и вывод их на печать; обработка гиперспектральных снимков; инструменты, позволяющие автоматизировать и сделать максимально удобной для пользователя комплексную обработку и анализ изображений (спектральный и топографический анализ, а также анализ растительности); удобный on-line help

(технические подсказки, с помощью ресурсов Интернета); использование формул и фильтров при обработке изображений.

В новой версии ENVI 5.3, вышедшей 11.08.2015, реализована возможность обработки новых каналов Landsat 8; данных лидара, получения новых индексов определения характеристик воды; осуществляется поддержка новых типов данных (тайловых продуктов Pleiades и SPOT, Google и др.); предусмотрено взаимодействие с новыми версиями 64-битных операционных систем (X Yosemite и Windows 10) и других ГИС- систем (ArcGIS 10.3); улучшена производительность программных приложений (обработка больше 50-ти изображений, увеличена скорость импорта и экспорта); добавлен просмотр профиля серии пространственно-временных растров, возможность создания большинства производных продуктов по облаку точек (например, ЦМР) и аналитических функций (например, линий прямой видимости) [4].

Приведенные характеристики демонстрируют быстрый отклик разработчиков программного обеспечения на появление новых сенсоров, необходимость увеличения производительности, скорости обработки, точности, одновременной совмещенной визуализации различных характеристик объектов.

Российская компания Esri CIS объявила о начале открытого бета-тестирования другого интересного продукта – Esri ArcGIS Earth, интегрированного в ArcGIS – нового бесплатного легкого настольного приложения для мгновенного просмотра 3D-карт. ArcGIS Earth может использоваться всеми сотрудниками организации в качестве простого и удобно инструмента для работы с картами и обмена информацией в различных форматах, включая KML и сервисы ArcGIS, как во внутрикорпоративной сети, так и через Интернет. Первый релиз ArcGIS Earth будет доступен для настольных компьютеров и планшетов на базе Windows; с последующим расширением поддерживаемых платформ. Система безопасности ArcGIS базируется на инфраструктуре безопасности предприятия или организации и предоставляет защищенный доступ для обмена картами и данными. ArcGIS Earth распространяется бесплатно для всех желающих.

Наиболее интересны возможности бесплатных открытых ГИС: SAGA, GRASS, ILWIS и Google Earth, QuantumGIS, gvSIG, JUMP.

Обратимся к системе SAGA [5]. SAGA изначально предназначалась для сбора, управление, анализа и представления данных для пространственного анализа и моделирования, аналитической визуализации. Ключевым свойством для их осуществления является взаимодействие с различными аппаратными базами, способы представления данных, пространственные характеристики. Являясь гибридной ГИС, SAGA поддерживает векторную и растровую модели данных, позволяет выполнять обработку «облаков точек», получаемых в результате лидарной съемки. Анализ данных ориентирован на статистический анализ пространственных и пространственно-временных данных. SAGA работает под ОС MS Windows,

Linux, MacOS X, FreeBSD, в установленном виде занимает менее 100 Мб дискового пространства, не требует формальной инсталляции, операции реализуются в виде модулей. В SAGA реализуется объектно-ориентированный анализ изображений (Object Based Image Analysis – OBIA), позволяющий выделять группы пикселей (объектов) на основе их схожести по яркости (цвету), форме, размеру, текстуре, в том числе, анализ ЦМР (форму поверхности, освещенность, видимость, количество тепла, комплексные топографические индексы, гидрологический анализ, моделирование, геостатистический анализ и др.)

Например, SAGA позволяет моделировать распределение поверхностного стока 9 различными способами, а для оценки угла наклона поверхности предоставляет 7 алгоритмов.

Ниже приведена сравнительная таблица (табл. 1) обобщенных характеристик систем SAGA, GRASS и ILWIS, которые дают представление о возможностях ПО и сложности его освоения.

Благодаря реализации большого числа алгоритмов пространственного анализа, SAGA – одна из наиболее сильных на сегодняшний день ГИС аналитической направленности. Дальнейшие перспективы развития связываются с совершенствованием API и GUI, доработкой взаимодействия с Java, R.

Публикации авторов и активных пользователей в академических изданиях, которые углубленно знакомят с реализованными в SAGA алгоритмами и освещают вопросы ее применения для решения прикладных задач, находятся в открытом доступе. Полезными могут быть и учебные материалы, посвященные статистическому анализу пространственных и пространственно-временных данных с применением бесплатного и открытого ПО – R, SAGA, GRASS, FWTools, Google Earth и прочего. Функциональные возможности SAGA детально рассматривались в рамках семинаров, начиная с 2011 года.

Т а б л и ц а 1

Сравнительная характеристика SAGA, GRASS и ILWIS [5]

ГИС	SAGA	GRASS	ILWIS
<i>Общие</i>			
Полное название	System for Automated Geoscientific Analyses	Geographic Resources Analysis Support System	<i>Integrated Land and Water Information System</i>
Координирующая организация	кафедра физ. географии, Институт географии, Гамбургский университет	OSGeo	<i>52°North Initiative for Geospatial Open Source Software GmbH</i>

Таблица 1 продолжение

Направленность	пространственный анализ и моделирование, аналитическая визуализация	управление данными, обработка изображений, пространственное моделирование,	анализ растровых данных, обработка изображений, картографическая визуализация
		аналитическая визуализация, картографическая визуализация	
Официальный веб-сайт	http://saga-gis.org/ [5]	http://grass.osgeo.org/ [6]	http://52north.org/communities/ilwis/ [7]
Актуальная версия	2.1.1 от 09.01.2014	6.4.3 от 30.07.2013	3.08.03 от 22.03.2013
Лицензия	LGPL для API, GPL	GPL	GPL
Язык программирования	C++	C, Tcl/Tk, Python	C
Платформы	MS Windows, Linux, MacOS X, FreeBSD	MS Windows, Linux, MacOS X, FreeBSD и многие другие	MS Windows, a Linux, Mac OS X – через приложение Wine
Импорт/ экспорт растровых/ векторных данных	GDAL/OGR (Geospatial Data Abstraction Library)	GDAL/OGR	GDAL и поддержка большинства векторных форматов (.e00, .lin, .pts ascii, .shp, .bna, .dxf, .smt, .seq)
Поддержка стандартов OGC	WMS, WFS	WMS, WFS, SFS, GML	WMS, WFS, SFS, GML
Чтение/ запись БД, GUI, стандартные возможности	+	+	+
Возможности расширения	Python, консольные скрипты	Shell, Bash, Python, Perl	собственный скриптовый язык ILWIS
Уровень пользователя	новичок – исследователь	опытный – исследователь	новичок - исследователь

Таблица 1 продолжение

Координатные трансформации/ проекции	PROJ.4, GeoTrans	PROJ.4	собственная библиотека, в т.ч. проекции для геостационарных спутников
Общее число функций (модулей)	более 600	более 400	более 100
Аналитические возможности			
Работа с данными LiDAR	+	+	–
Подготовка ДДЗ (фильтрация, орто-ректификация)	+	+	+
Анализ изображений:			
классификация с обучением и без	+	+	+
объектно-ориентированный анализ изображений	+	±	–
Анализ ЦМР			
форма поверхности	+	+	+
освещенность, видимость, количество тепла	+	+	+
комплексные топографические индексы	+	+	+
гидрологический анализ	+	+	+
Геостатистика	+	через интерфейс R-GRASS	+
Моделирование	±	+	–

Следующая система, представляющая интерес – свободная географическая информационная система с открытым кодом QGIS [8], созданная сообществом свободного программного обеспечения с открытым кодом (FOSS), распространяющаяся под GNU General Public License [9]. О применении лицензии GNU для собственных программ можно узнать на сайте [10].

Она работает на Linux, Unix, Mac OSX, Windows и Android; поддерживает большинство векторных и растровых (GeoTIFF, ERDAS IMG, ArcInfo ASCII GRID, JPEG, PNG и др.) форматов; интегрируется с GRASS GIS (можно просматривать и редактировать слои GRASS); позволяет осуществлять просмотр данных из баз данных (например, Oracle, PostGIS, MS SQL Server); использовать язык баз данных SQL, исследовать, управлять и анализировать данные; создавать карты; собирать многоканальные косми-

ческие изображения (Landsat), обрабатывать данные лидара; дополнять модули, используя C++ или Python, настраивать модули под пользователя.

QGIS включает модули: «захват координат», «соединение с БД», «преобразователь Dxf2Shp», «обозреватель событий», инструменты анализа, выборки, геопроецирования, интерполяции, редактирования, «теплокарта», «пространственные запросы», «проверка топологии», «зональная статистика» и др.

Курс лекций, читаемых на кафедре дистанционного зондирования и инвентаризации лесов Геттингенского университета, включает знакомство с QGIS [11]. Калифорнийский университет департамента биологии и сельского хозяйства использует связку QGIS – GRASS – R для анализа пространственных данных в экологии и сельском хозяйстве.

Таким образом, открытые продукты позволяют удовлетворить большинство потребностей решения прикладных задач.

Другой составляющей освоения ГИС-технологий является возможность открытого доступа к изображениям местности. Компания «Сканэкс» знакомит с результатами тематической обработки спутниковых данных, принимаемых собственными станциями Унискан ТМ в интересах народного хозяйства РФ, представляя их на своих тематических сервисах, например, сервисе «Карта пожаров». Команда «Карты пожаров» подвела итоги работы в 2015 году и выделила наиболее значимые, с точки зрения развития сервиса, события. Среди них — запуск международной версии firesmap.com и расширение перечня источников спутниковых данных [12]. К сервису обращались не только российские пользователи, но и жители Беларуси, Украины и Казахстана. Создание международной версии «Карты пожаров» естественным образом было связано с расширением перечня источников спутниковых данных. В частности, в рамках сотрудничества компании «СКАНЭКС» с американским Университетом Висконсин-Мэдисон была организована интеграция в «Карту пожаров» данных со спутника Suomi NPP, покрывающих территорию Северной Америки. Новый источник спутниковых данных, в дополнение к глобальным данным сервиса FIRMS (NASA), обеспечит более оперативное покрытие указанной территории.

Еще одним новым источником данных стала глобальная мозаика MODIS (слой «GIBS mosaic Terra/Aqua»). Для доступа к этому сервису разработчиками был реализован плагин для Leaflet (сервис GIBS предоставляет доступ к целому перечню продуктов на основе съемки MODIS), который выложен в публичном репозитории [13]. Сервис предоставляет доступ ко всем продуктам в виде растровых тайлов, а значит легко интегрируется во все широко используемые веб-приложения.

Помимо доступа к информации о пожарах, публикуемой на сайте, разработчики внешних информационных систем, используя API «Карты пожаров», могут получить доступ ко всем источникам, включая оперативные данные сети «СКАНЭКС», через единый программный сервис [14].

Архитектура автономной системы мониторинга пожаров приведена на рис.1 [12], а пример карты пожаров 2015 года – на рис. 2.

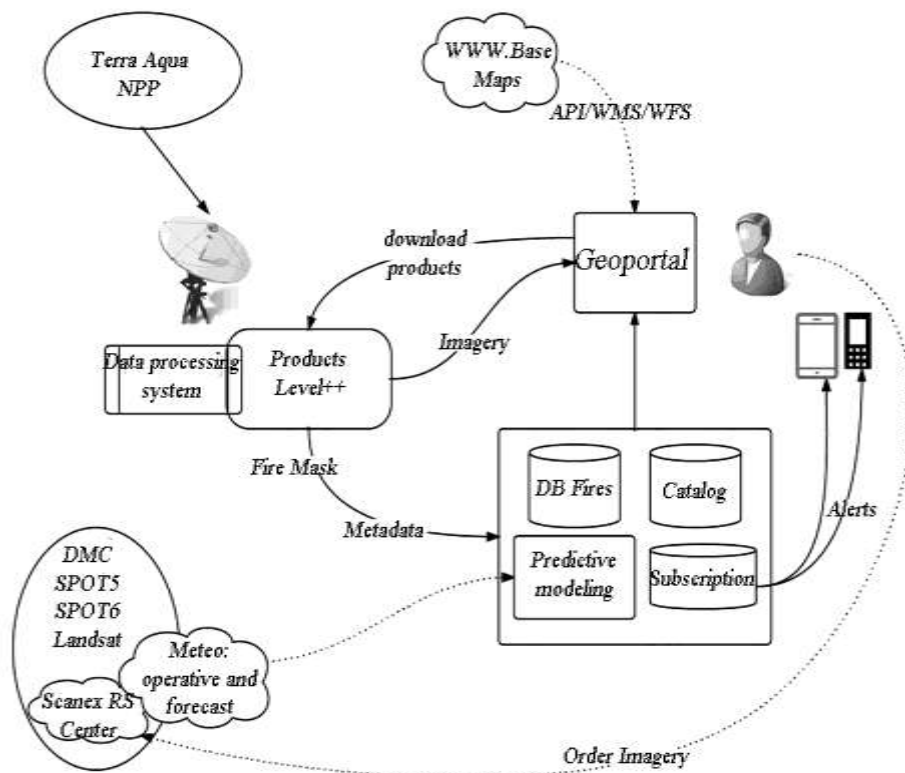


Рис.1. Архитектура автономной системы мониторинга пожаров

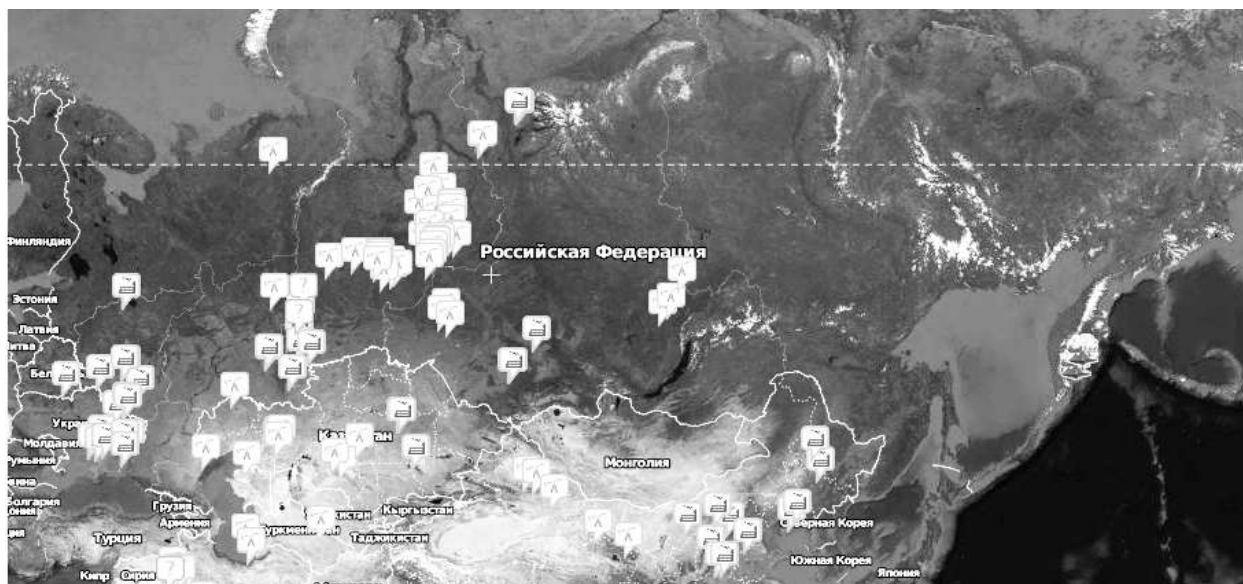


Рис.2. Карта пожаров 2015 года

Заключение

Использование спутниковых данных в разных отраслях народного хозяйства, в том числе, лесного хозяйства увеличивается с каждым годом.

Для распространения передовых мировых технологий необходимо повышать уровень обучения ГИС-технологиям, используя возможности открытого программного обеспечения и Фонда свободного программного обеспечения, предоставляющего право получения лицензий на использование и дополнение вновь разработанными модулями используемых систем, а также данные открытых сайтов ведущих компаний–разработчиков.

Библиографический список

1. <http://www.erdas.com>
2. <http://www.gisa.ru/1506.html>
3. <http://sovzond.center/>
4. www.enviofficial.com
5. <http://saga-gis.org>
6. <http://grass.osgeo.org/>
7. <http://52north.org/communities/ilwis/>
8. <http://pro-spo.ru/-linux-e/2063--qgis>
9. <http://www.gnu.com>
10. <http://www.gnu.org/licenses>.
11. http://wiki.gis-lab.info/w/Учебник_Quantum_GIS
12. <http://www.scanex.ru/>
13. <https://github.com/aparshin/leaflet-GIBS>
14. <http://fires.ru/help.html#enterprise>

ОГЛАВЛЕНИЕ

Гурову Сергею Владимировичу Преснякову Владимиру Александровичу посвящается.....	3
А.М.Заяц. Итоги научно-исследовательской работы и НИРС 2016 год.....	5
А.М.Заяц. Обзор беспроводных сенсорных сетей и технологий информационных систем оценки лесной пожароопасности и мониторинга лесов.....	9
И.В.Панфилов. Информационная технология защиты информации в компьютерных системах.....	21
С.П.Хабаров. Терминальный доступ между UBUNTU и WINDOWS компьютерами.....	30
С.П.Хабаров, В.М.Амбросовский, А.С.Корнев. Обоснование выбора интерфейсов в системе координированного управления с использованием групповой экспертной оценки.....	43
Н.П.Васильев. Конечные автоматы для задач расшифровки и проверки научных названий растений.....	49
Ю.А.Жук. Классификация уровня знаний студентов на основе методов машинного обучения.....	55
Н.В.Лушкин. Реализация алгоритма определения чисел бетти в графических объектах.....	61
М.О.Лебедев. Создание клиентской приложений для информационных систем со сложной структурой данных.....	70
И.А.Обухова, Р.М.Яковлев. Радиационная безопасность в околоземном пространстве и в космосе. Результаты расчетов.....	77
Н.Г. Полетаева, Д.О. Туктарев. Защита данных криптографическими средствами СУБД ORACLE.....	87
С.П.Хабаров, Е.И.Шалаев, С.П. Васильев. Реализация модуля логического вывода в составе информационной системы классификации растений.....	99
М.Л.Шилкина. Применение различных видов преобразований координат для коррекции картографических изображений.....	109
М.А.Шубина. Новые программные продукты ГИС-технологий.....	118

Научное издание

Отв. редактор
Зяц Анатолий Моисеевич

**ИНФОРМАЦИОННЫЕ
СИСТЕМЫ И ТЕХНОЛОГИИ:
ТЕОРИЯ И ПРАКТИКА**

Сборник научных трудов

Выпуск 8

Отпечатано в авторской редакции с готового оригинал-макета

Подписано в печать с оригинал-макета 25.03.16.
Формат 60×84/16. Бумага офсетная. Печать трафаретная.
Уч.-изд. л. 8,0. Печ. л. 8,0. Тираж 100 экз. Заказ № 47. С 7.

Санкт-Петербургский государственный лесотехнический университет
Издательско-полиграфический отдел СПбГЛТУ

194021, Санкт-Петербург, Институтский пер., 3.