

**Министерство образования и науки Российской Федерации
ГОУ ВПО «Магнитогорский государственный университет»**

**СТАНДАРТИЗАЦИЯ В ОБЛАСТИ ИНФОРМАЦИОННЫХ
ТЕХНОЛОГИЙ**

**СБОРНИК ТРУДОВ УЧАСТНИКОВ СЕМИНАРОВ
ЛАБОРАТОРИИ ОТКРЫТЫХ СИСТЕМ**

Магнитогорск
2009

ББК 397
УДК 65.0:004

С Стандартизация в области информационных технологий: сборник трудов участников семинаров лаборатории открытых систем. - Магнитогорск: МаГУ, 2009. – 118 с.

Научные редакторы: А.Я. Олейников, А.В. Меркулова

В сборнике представлены доклады участников семинаров лаборатории открытых систем: «E-learning: технологии, стандарты, перспективы», «Технологии Opensource», «Разработка и стандартизация программных средств и информационных технологий». Доклады отражают результаты научно-исследовательской и научно-методической работы ведущих специалистов в области информационных технологий из разных отраслей промышленности, науки и высшего образования.

ISBN

© Магнитогорский
государственный университет, 2009
© Авторы публикаций

Содержание

Раздел I. Разработка и стандартизация программных средств и информационных технологий	5
Батоврин В.К., Олейников А.Я., Широбокова Т.Д. ОБЕСПЕЧЕНИЕ ИНТЕРОПЕРАБЕЛЬНОСТИ ПРИ СОЗДАНИИ ЭЛЕКТРОННОГО ГОСУДАРСТВА.....	5
Каменщиков А.А. СТАНДАРТИЗАЦИЯ В МЕДИЦИНЕ НА ПРИМЕРЕ ПРОЕКТА НЕНТА.....	10
Меркулова А.В., Редькин Д.П., Скрыбина Т.Л. ЛИЦЕНЗИРОВАНИЕ ПРОГРАММНЫХ СРЕДСТВ.....	16
Петеляк В.Е. ВЛИЯНИЕ ОТКРЫТОСТИ КРИПТОГРАФИЧЕСКИХ СТАНДАРТОВ НА ИХ ПРИМЕНЕНИЕ.....	19
Раздел II. E-learning: технологии, стандарты, перспективы	26
Лактионова Ю.С. ОСНОВНЫЕ АСПЕКТЫ И ВОЗМОЖНОСТИ ТЕОРИИ РАЗВИВАЮЩЕГО ОБУЧЕНИЯ И ТЕХНОЛОГИИ E-LEARNING ДЛЯ ПОВЫШЕНИЯ ЭФФЕКТИВНОСТИ ОБУЧЕНИЯ ИНФОРМАТИКЕ.....	26
В.Б. Лапшина, В.Н. Макашова РАЗВИТИЕ РЫНКА E-LEARNING УСЛУГ В РОССИИ.....	32
Махмутова М.В. АСПЕКТЫ ИНТЕГРАЦИИ ТРАДИЦИОННОЙ И ДИСТАНЦИОННОЙ ТЕХНОЛОГИЙ ОБУЧЕНИЯ В ВУЗЕ.....	37
Меркулова А.В. E-LEARNING: СУЩНОСТЬ ПРОБЛЕМЫ ВНЕДРЕНИЯ.....	43
Рубан К.С. ИНТЕРОПЕРАБЕЛЬНОСТЬ КОМПОНЕНТОВ ЭЛЕКТРОННОГО ОБУЧЕНИЯ.....	48
Чернова Е.В. РИСКИ ДИСТАНЦИОННОГО ОБРАЗОВАНИЯ.....	54
Чудинова Ю.А., Шестов М.А. ИСПОЛЬЗОВАНИЕ МУЛЬТИМЕДИЙНЫХ ОБУЧАЮЩИХ СИСТЕМ ДЛЯ ПОДГОТОВКИ ТЕХНОЛОГИЧЕСКОГО ПЕРСОНАЛА ПРОМЫШЛЕННЫХ ПРЕДПРИЯТИЙ НА ПРИМЕРЕ ОАО «ММК».....	58
Раздел III. Технологии Opensource	64
А.А. Бармин СППР ВЫБОРА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ ВЕБ- СЕРВЕРА.....	64

Г.С. Канин	69
СТАТИСТИЧЕСКИЙ АНАЛИЗ В ПРОГРАММНОМ СРЕДСТВЕ GNUMERIC.....	
Г.А. Лисьев, В.Г. Измайлов	
ИНСТРУМЕНАЛЬНАЯ ПОДДЕРЖКА НАУЧНЫХ ИССЛЕДОВАНИЙ И ОТКРЫТОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ.....	78
А.В. Меркулова	
К ВОПРОСУ ОБ ИСПОЛЬЗОВАНИИ ТЕХНОЛОГИЙ OPENSOURCE.....	82
И.В. Попова, С.Ю. Давлеткулов	
ЭКОНОМИЧЕСКИЕ АСПЕКТЫ OPEN SOURCE.....	84
И.В. Попова, В.В. Мурлаева	
СРАВНИТЕЛЬНЫЙ АНАЛИЗ СРЕДСТВ МОДЕЛИРОВАНИЯ DIA И MS VISIO.....	86
И.В. Попова, А.В. Яриз	
ОБЗОР СВОБОДНО РАСПРОСТРАНЯЕМЫХ СРЕД ЛОГИЧЕСКОГО ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ ПРОЛОГ...	91
И. В.Портнов	
ФАЙЛОВАЯ СИСТЕМА VTRFS: РЕАЛЬНОСТЬ И ПЕРСПЕКТИВЫ.....	96
И. В.Портнов, М.Сорокин	
РАЗРАБОТКА JAVA ПРИЛОЖЕНИЙ В NETBEANS.....	103
К.А. Рубан	
ПРИМЕНЕНИЕ СИСТЕМ ОБНАРУЖЕНИЯ ВТОРЖЕНИЙ НА ПРИМЕРЕ ПРОГРАММЫ TRIPWIRE.....	107
С. Шакшин	
О ПРОЕКТЕ CENTRIX.....	113

Раздел I. Разработка и стандартизация программных средств и информационных технологий

В.К.Батоврин
Московский государственный институт радиотехники, электроники и
автоматики (технический университет)
г.Москва

А.Я. Олейников
Учреждение Российской академии наук Институт радиотехники и
электроники им. В.А. Котельникова РАН
г.Москва

Т.Д.Широбокова
Учреждение Российской академии наук Институт радиотехники и
электроники им. В.А. Котельникова РАН
г.Москва

ОБЕСПЕЧЕНИЕ ИНТЕРОПЕРАБЕЛЬНОСТИ ПРИ СОЗДАНИИ ЭЛЕКТРОННОГО ГОСУДАРСТВА

В настоящее время все больше количество государств в Америке, Европе и Азии, включая Новозеландию занимается построением электронного государства. В нашей стране эти работы ведутся в рамках ФЦП «Электронная Россия». В настоящее время предстоит разработать новую программу - "Электронное Правительство" до 2015 года (<http://www.government.ru/content/governmentactivity/mainnews/archive/2009/05/27/7008326.htm>). Под электронным государством (правительством) принято понимать способ осуществления информационных аспектов государственной деятельности, основанный на использовании ИКТ-систем. Во всех развитых странах электронное государство является инструментом достижения целей административной реформы, позволяющим повысить доступность государственных услуг. В отношении этого показателя в Стратегии развития информационного общества в РФ определено, что к 2015 году уровень доступности для населения базовых услуг в сфере информационных и телекоммуникационных технологий должен составить сто процентов.

Основным технологическим инструментом обеспечения доступности услуг, предоставляемых ИКТ-системами, является придание этим системам свойства интероперабельности, т.е. обеспечение способности двух или более систем или их элементов к обмену информацией и к эффективному использованию информации, полученной в результате обмена. Будучи одним из базовых признаков открытых систем интероперабельность может проявляться в различной форме. Соответственно различают техническую, синтаксическую, семантическую, организационную, прагматическую, федеративную и другие виды интероперабельности, которые в различной степени должны быть обеспечены при создании систем электронного государства.

В основе предлагаемой нами методики обеспечения интероперабельности систем электронного государства лежит выработка и принятие решений на основе системного анализа с использованием принципов открытых систем и архитектурного подхода.

Основной принцип открытых систем заключается в создании среды, которая включает всеобъемлющий набор интерфейсов, служб и поддерживаемых форматов, а также представлений пользователя об интероперабельности или переносимости приложений, данных или людей, реализованных в соответствии со стандартами и профилями информационных технологий.

Это предполагает:

- выделение ключевых интерфейсов и определение связанных с ними служб и форматов, а также представлений пользователей;
- реализацию решений на основе развивающихся, доступных и общепризнанных открытых стандартов;
- построение профиля – согласованного набора стандартов и международных функциональных стандартов, необходимых для выполнения определённой функции или набора функций.

В рамках архитектурного подхода формируется совокупность архитектурных описаний электронного государства, базирующаяся на целостном многоаспектном представлении о создаваемой системе и обеспечивающая благодаря этому эффективные возможности её развития, естественной интеграции в среду функционирующего предприятия и интероперабельности с взаимосвязанными с нею системами.

Как показывает анализ, подобная методика развивается в рамках ряда инициатив по созданию электронного государства, в частности в программе Евросоюза *Interoperable Delivery of European e-Government Services to public Administrations, Businesses and Citizens (IDABC)* и при разработке европейской стратегии интероперабельности *European Interoperability Strategy – EIS*. В результате строится модель, которая после одобрения и принятия становится основным инструментом для развития работ по обеспечению интероперабельности в среде электронного государства. Примеры подобных моделей можно найти в работах [1,2].

На современном этапе развития электронного государства основное внимание при моделировании следует уделять обеспечению высокоуровневой не технической интероперабельности. Это предполагает согласование целей, ролей и политик, относящихся к узловым вопросам работы и развития системы электронного государства, на основе модели функционирования, с обеспечением способности различных бизнес-субъектов, бизнес-объектов и бизнес-процессов, использующих возможно различную информационную инфраструктуру, к согласованному функционированию на основе обмена информацией и формирования согласованных потоков работ и набора услуг, предоставляемых потребителям, а также механизма идентификации и оценки приоритета таких услуг на различных уровнях организации. Построение такой модели требует больших затрат времени и поиска компромисса между большим числом заинтересованных сторон.

Для решения указанной задачи к работам следует активно привлекать специалистов из области общественных наук, в частности, Евросоюз при разработке и развитии моделей интероперабельности привлекает таких специалистов к проведению специальных опросов граждан – потенциальных потребителей услуг электронного государства.

При моделировании интероперабельности возможны два подхода:

– *материализованная координация (warehoused enterprise)* – создание новой сущности, в которой обеспечен необходимый уровень согласования для заранее определенных потоков работ, процессов и архитектур;

– *виртуальная координация (virtual enterprise)* – необходимый уровень согласования устанавливается по мере необходимости, с возможностью изменения набора затрагиваемых потоков работ, процессов и архитектур.

Сегодня в нашей стране преимущественно используется первый подход, тогда как большая часть успешных зарубежных проектов реализует, и на наш взгляд обоснованно, второй подход.

Модели обеспечения интероперабельности основанные на виртуализации, могут строиться в следующих вариантах:

– ***Иерархическая модель***, с выделением ответственной организации, задающей иерархическую структуру потоков работ на основе одного из трех подходов:

- *Централизованного*, когда доминирующее или специально назначенное ведомство полностью определяет общую структуру потока работ;

- *С участием исполнителей (participative)*, когда все заинтересованные стороны вовлекаются в процесс определения структуры потока работ;

- *Децентрализованного*, когда каждая из заинтересованных сторон независимо формирует свою часть общего потока работ.

– *Рыночная модель*, основанная на предположении, что формальные соглашения между заинтересованными сторонами не заключаются, каждая из организаций, ответственных за определенный поток работ, выбирает сервис-провайдера, включая интерфейс, используемый для обеспечения потоков работ и обмена данными.

– *Ситуационная (ad-hoc) модель*, основанная на предположении, что предварительное определение структуры потока работ не проводится и процессы выполняются согласно пожеланиям организаций с учетом их текущих потребностей.

Независимо от принятого подхода к моделированию, можно выделить четыре группы ключевых факторов успеха, относящихся к обеспечению не технической интероперабельности, а именно:

– *политические*, связанные с основными политическими и институциональными проблемами в привязке к политическим деятелям;

– *законодательные*, связанные с потребностью в законодательных актах;

– *управленческие*, связанные с организационными, управленческими и технологическими проблемами и их решением путем обращения к чиновникам и управляющим;

– *экономические*, связанные с финансированием и бюджетированием.

Выводы:

1. Построение системы электронного государства невозможно без решения проблемы интероперабельности и разработки соответствующих моделей и стандартов.

2. Однажды принятая модель становится фактическим стандартом на длительный период времени и её замена потребует очень больших затрат.

3. Обеспечение технической интероперабельности составляет только часть (причем, не самую большую) задачи и здесь имеются приемлемые модели.

4. Сегодня наибольшее значение приобретают проблемы обеспечения нетехнической интероперабельности, успешное решение которых возможно только при согласованной работе управленцев и инженеров.

5. Необходимо разработать Рекомендации по обеспечению интероперабельности в среде электронного государства и принять их на государственном уровне. Эти рекомендации могут служить основой обеспечения интероперабельности и в системах меньшего масштаба, например, электронного города или электронной губернии. Для выработки таких Рекомендаций обязательно должен быть сформирован уполномоченный орган на каждом уровне управления.

Литература:

1. Батоврин, В.К., Гуляев, Ю.В., Олейников, А.Я. Методы и средства обеспечения интероперабельности/В.К. Батоврин, Ю.В. Гуляев, А.Я. Олейников//Стандартизация информационных технологий и интероперабельность: Труды второй международной конференции SITOP 2008. – М.: Экспоцентр, 2008. – с. 14-17.

2. European Interoperability Framework for Pan-European e-government services. Draft document as basis for EIF 2.0 [Электронный ресурс] – Режим доступа: <http://ec.europa.eu/idabc/servlets/Doc?id=31597>.

А.А. Каменщиков
Учреждение Российской академии наук Институт радиотехники и
электроники им. В.А. Котельникова РАН
г.Москва

СТАНДАРТИЗАЦИЯ В МЕДИЦИНЕ НА ПРИМЕРЕ ПРОЕКТА НЕНТА¹

Сегодня все большую значимость приобретают вопросы стандартизации информационных технологий при создании Единого

¹ Работа выполнена при поддержке Российского фонда фундаментальных исследований, грант № 09-07-00171-а

информационного пространства (ЕИП) в здравоохранении. При создании ЕИП одной из наиболее важных проблем выступает проблема интероперабельности (возможности взаимодействия) как внутри, так и между лечебно-профилактическими медицинскими учреждениями. Обеспечение интероперабельности направлено, в первую очередь на то, чтобы избежать дублирования данных [1].

Представляется крайне важным, чтобы обеспечение интероперабельности составляло часть национальной политики в здравоохранении. В этом отношении хорошим примером может служить австралийский проект «The National E-Health Transition Authority» (НЕНТА). Данный проект динамично развивается, все достижения размещаются на сайте в открытом доступе. Одной из основных целей данного проекта является развитие интероперабельности в медицине. Ключ достижения интероперабельности – открытость и популяризация информации по ее достижению.

В проекте НЕНТА под понятием интероперабельность используется следующее определение: Интероперабельность – это способность систем обмениваться информацией и/или взаимодействовать с другой системой, опираясь на общие стандарты [2].

В НЕНТА разработана структура интероперабельности е-здоровья(e-health), согласованная на всех уровнях с правительством Австралии. Будущие системы е-здоровья, принимаемые австралийским правительствам для лечебных учреждений страны должны демонстрировать их согласованность с архитектурой, которая в свою очередь определяется основным документом «НЕНТА interoperability framework» (<http://www.nehta.gov.au/>). Данный документ:

- Устанавливает общие понятия е-здоровья, принципы и стандарты, чтобы продвигать и делать возможной интероперабельность технологий, информации здравоохранения, организационного уровня;

- Гарантирует, что в будущих системах е-здоровья смогут предсказуемо изменяться с учетом изменений в технологиях, информации здравоохранения, на организационном уровне;

- Обращается к соответствию систем е-здоровья с требованиями интероперабельности НЕНТА.

НЕНТА в настоящее время имеет несколько инициатив в стадии реализации, чтобы обеспечить безопасные, интероперабельные системы е-здоровья, многие из которых очень уместны, и сильно зависимы от структуры документа «Interoperability Framework», который, в частности, обеспечивает построение таксономии стандартов здравоохранения.

Данные инициативы заключаются в:

- Урегулирование стандартов для, кодирования первоочередной информации здравоохранения – например: направления, отчеты больницы - записи системами е-здоровья;

- Установление стандартных клинических терминов для использования в системах е-здоровья, так, чтобы системы использовали не противоречащие термины, для описания одной и той же болезни, терапии, медикамента и т.д.;

- Идентификация безопасного средства электронной передачи информации здоровья между уполномоченными профессионалами здравоохранения;

- Проектирование спецификаций для безопасных электронных отчетов здоровья, чтобы позволить уполномоченным профессионалам здравоохранения, просматривать совокупную историю болезни человека, поддерживая высокие стандарты секретности;

- Развитие уникальных идентификаторов для людей, поставщиков здравоохранения и медицинских продуктов, так, чтобы правильная информация была назначена для соответствующего пациента;

- Развитие механизма для вовлечения национальных/международных стандартизирующих организаций, чтобы поддержать выполнение работы НЕНТА.

Возвращаясь к документу НЕНТА «Interoperability Framework», важно подчеркнуть, что он использует модель, построенную на основе модели - Capability Maturity Model Integration (CMMI). Interoperability Maturity Model (IMM), выделяя особенности для организаций е-здоровья:

- определяет 5 уровней зрелости (maturity levels) на основании CMMI;
- устанавливает цели интероперабельности (interoperability goals) с точки зрения важности для страны;

Существует множество уровней интероперабельности, но всегда можно выделить три основных уровня [3]:

- технический уровень;
- семантический уровень;
- организационный уровень;

Остальные уровни, по сути, будут являться расширением данных трех уровней, в зависимости от задач возникает необходимость в детализации того или иного уровня. На рисунке 1, показаны 5 уровней, которые выделяют разработчики в НЕНТА. В данном случае расширен технический уровень интероперабельности, из которого еще выделяют: синтаксический и структурный уровень [3]. В красных блоках показаны основные объекты, которые рассматриваются на каждом из уровней, в желтых блоках – механизмы, стандарты, структуры, с помощью которых достигается интероперабельность на соответствующих уровнях. Основным механизмом обеспечения интероперабельности – это использование стандартов на каждом из уровней.

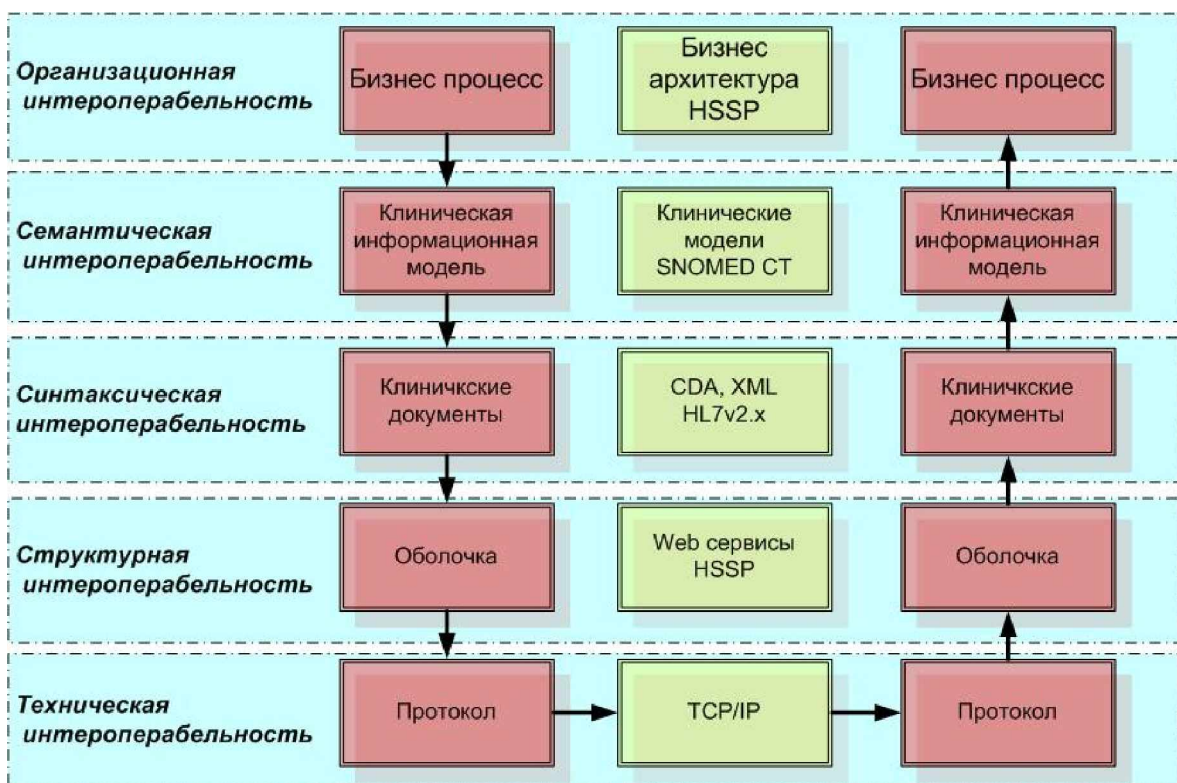


Рисунок 1 – Уровни интероперабельности

НЕНТА ведет свой каталог стандартов, в котором стандарты классифицированы по двум схемам:

- Первая часть каталога разбивает стандарты на основании 8 основных направлений работ НЕНТА:

1. Clinical Communications
2. E-Health Interoperability
3. Engagement and Adoption
4. Identity Management
5. Secure Messaging
6. Shared Electronic Health Record
7. Supply Chain
8. Unique Healthcare Identification

- Вторая часть каталога разбивает стандарты на 3 части с точки зрения интероперабельности:

1. Informational
2. Organisational
3. Technical

Однако такой подход классификации стандартов не может полностью охватить весь спектр стандартов, разнообразных для e-здоровья. Также существуют стандарты, которые относятся к нескольким уровням интероперабельности, в связи с этим возникает необходимость созданию описаний по взаимодействию стандартов. Данной работой активно занимается организация Integrating the Healthcare Enterprise (<http://www.ihe.net>).

Следует отметить, что по данным НЕНТА интероперабельность у многих вызывает опасения из-за определенных трудностей при ее достижении. С точки зрения НЕНТА интероперабельность воспринимается слишком сложной, т.к.:

- Требуется большой объем дополнительных работ, выгоды от которых не проявляются быстро;
- Ожидаются более высоких затрат;
- Требуются серьезные изменения в организации работ на разных уровнях;
- Подчеркивается необходимость эффективной архитектуры государства;
- Нехватка компетентных руководителей и динамичного подхода развития;
- Расценивается, как техническая проблема и путают с интеграцией;
- Отрицание риска укрепляет изолированные решения.

Данные опасения разделяют большинство разработчиков медицинских информационных систем (МИС) России. Однако развитие интероперабельности МИС является крайне важной задачей.

Более подробную информацию о проекте НЕНТА можно найти на сайте проекта <http://www.nehta.gov.au/>. Обсуждение вопросов интероперабельности для медицины доступно на сайте [www](http://www.ithealth.ru/) <http://www.ithealth.ru/>. Актуальные материалы о стандартизации в

здравоохранении России наиболее полно изложены в интервью А.П. Столбова [4]

Литература:

1. Гончаров, Н.Г., Каменщиков, А.А. Проблемы интероперабельности в медицине/ Н.Г.Гончаров, А.А.Каменщиков//Труды первой международной конференции «Стандартизация информационных технологий и интероперабельность». – М.: Экспоцентр, 2007. – с. 83
2. Andy Bond Dr. The Path for Interoperability [Электронный ресурс]/Andy Bond. – 2007. – Режим доступа: http://www.nehta.gov.au/index.php?option=com_docman&task=doc_download&gid=267&Itemid=139.
3. Andy Bond HL7 Interop Auckland Sept 2007 [Электронный ресурс]/Andy Bond. – 2007. – Режим доступа: http://www.nehta.gov.au/index.php?option=com_docman&task=doc_download&gid=385&Itemid=139.
4. О недостатках нынешней процедуры рассмотрения проектов стандартов в интервью заместителя директора МИАЦ РАМН А.П. Столбова, которое он дал агентству [Электронный ресурс]. – Режим доступа: <http://www.cnews.ru/reviews/free/publichealth/int/miac/>

А.В. Меркулова
ГОУ ВПО «Магнитогорский государственный университет»
г.Магнитогорск

Д.П. Редькин
ГОУ ВПО «Магнитогорский государственный университет»
г.Магнитогорск

Т.Л. Скрябина
ГОУ ВПО «Магнитогорский государственный университет»
г.Магнитогорск

ЛИЦЕНЗИРОВАНИЕ ПРОГРАММНЫХ СРЕДСТВ

Отказ от использования нелегального программного обеспечения — такая тема занимает в последнее время умы едва ли не большинства ИТ-

директоров. Главным движущим фактором является усиление контроля в правовой сфере, громкие уголовные процессы, изменения в законодательстве в сфере защиты авторских и смежных прав, участившиеся случаи привлечения к ответственности ИТ-директоров и системных администраторов. При этом плюсы перехода на лицензионное программное обеспечение (ПО) очевидны: это доступность технической поддержки, своевременные обновления безопасности.

Использование лицензионного программного обеспечения – это гарантия юридической ответственности, корректной и стабильной работы программы, наличие документации и технической поддержки, безопасности в сети и отсутствия утечки информации.

Многие производители программного обеспечения предлагают специальные «программы лицензирования» для корпоративных заказчиков, смысл которых сводится к тому, что при приобретении большего количества лицензий цена за одну лицензию становится меньше. Для предприятий и организаций программное обеспечение часто продается пакетами на 5, 10, 25 и более рабочих мест.

Согласно статье 2 Закона о лицензировании лицензией признается специальное разрешение на осуществление конкретного вида деятельности при обязательном соблюдении лицензионных требований и условий, выданное лицензирующим органом юридическому лицу или индивидуальному предпринимателю.

Как видно из данного определения соискателем лицензии могут быть только действующие юридические лица и индивидуальные предприниматели.

Под понятием лицензируемой деятельности следует понимать вид деятельности, на осуществление которой на территории Российской Федерации требуется получение лицензии. То есть само наличие лицензии является для организации необходимым условием на право заниматься подобной деятельностью. Подобный вывод закреплен также в пункте 3

статьи 49 ГК РФ, которая прямо предусматривает, что возможность юридического лица осуществлять деятельность, на занятие которой необходимо получение лицензии, возникает с момента получения такой лицензии или в указанный в ней срок и прекращается по истечении срока ее действия, если иное не установлено законом или иными правовыми актами.

Лицензирование программного обеспечения (ПО) - это процесс, направленный на легализацию или попросту это покупка законных прав на использование программного обеспечения, которое установлено в компании. Лицензирование программного обеспечения занимает определенное время, но получить права на использование его для деятельности значит многое. Прежде всего - это гарантия безопасности и надежности бизнеса.

Программное обеспечение - это продукт интеллектуальной собственности, который имеет основную защиту от копирования, изменения и не законного использования в виде лицензии. Получить лицензию значит согласиться с правилами его использования прописанных в договоре о лицензировании программного обеспечения. И любое нарушение или отступление от этого договора лицензирования влечет за собой уголовную ответственность или ответственность, прописанную в этом договоре. Поэтому при незаконном использовании каких-либо программных продуктов, будь-то операционная система или же какое-то прикладное приложение, нужно помнить, что это незаконная деятельность.

При построении серьезных интеллектуальных инфраструктур в бизнесе, использование незаконного программного обеспечения нецелесообразно, по многим причинам. Главными из которых можно выделить финансовые затраты и временное бремя, связанное с решением юридических вопросов при оперативной проверке компании на предмет обнаружения незаконного ПО.

Основные цели лицензирования программного обеспечения, которое используется в компании:

1. Гарантия безопасности и надежности работы ПО.

2. Получение законных прав на использование программного продукта в своих целях.

3. Получение доступа к бесплатным обновлениям, сервисам и ресурсам, а также получение бесплатных консультаций по использованию лицензионного программного продукта.

Уже давно отошли те времена, когда было актуально и прибыльно держать свою бизнес деятельность на "пиратском" или незаконном программном обеспечении. Сегодня же все изменилось. Главным признаком серьезного и надежного бизнеса как раз стало лицензионное ПО.

Использование лицензионного программного обеспечения, показывает серьезность намерений не только клиентам или партнерам, работающих с компанией, но и основным конкурентам. Также, его использование, ведет к ограничению бизнеса от всевозможных рисков, связанных с ответственностью за незаконное использование данных программных продуктов.

В.Е. Петеляк
ГОУ ВПО «Магнитогорский государственный университет»
г.Магнитогорск

ВЛИЯНИЕ ОТКРЫТОСТИ КРИПТОГРАФИЧЕСКИХ СТАНДАРТОВ НА ИХ ПРИМЕНЕНИЕ

Введение

Долгое время криптографическое обеспечение использовалось преимущественно в военной и дипломатической деятельности и требовало значительных затрат, теоретические и практические аспекты криптографии были доступны ограниченному кругу лиц. С развитием и доступностью каналов связи и вычислительной техники, с увеличением объема гражданской информации, требующей защищенности, и со значительным увеличением числа пользователей такой информации, криптографическое

обеспечение получило широкое распространение как очень надежный и недорогой способ защиты информации и обеспечения безопасности действий в сети. Для гарантирования успешности таких коммуникаций необходима стандартизация криптографических алгоритмов и протоколов.

Криптографические стандарты

Под криптографическим стандартом в данной статье понимается нормативно-технический документ, устанавливающий характеристики криптографического алгоритма, условия и способы применения криптографического алгоритма и утвержденный компетентным органом.

Во многих странах криптография и связанная с ней деятельность подвергается особому регулированию: в США криптография приравнивается к вооружению и экспорт сильной криптографии запрещен; в России деятельность, связанная с криптографией, регулируется федеральным законом N 128-ФЗ «О лицензировании отдельных видов деятельности» [1], а используемые криптографические алгоритмы до недавнего времени предназначались только для служебного пользования. Следствием подобных ограничений является незначительное количество криптографических стандартов.

На национальном уровне выделим две группы криптографических стандартов:

1. стандарты США:
 - FIPS 197. Advanced Encryption Standard (AES) [2].
 - FIPS 186-2. Digital Signature Standard (DSS) [3].
 - FIPS 180-3. Secure Hash Standard (SHS) [4].
 - FIPS 46-3. Data Encryption Standard (DES) [5] (уже не действует, но сам криптографический алгоритм DES и его модификация 3DES до сих пор широко используются).
2. стандарты РФ:

- ГОСТ 28147-89 «Системы обработки информации. Защита криптографическая. Алгоритм криптографического преобразования» [6].
- ГОСТ Р 34.11-94 «Информационная технология. Криптографическая защита информации. Функция хэширования» [7].
- ГОСТ Р 34.10-2001 «Информационная технология. Криптографическая защита информации. Процессы формирования и проверки электронной цифровой подписи» [8].

Обе группы стандартов описывают одинаковый набор криптографических алгоритмов (симметричное шифрование, электронная подпись, функция хэширования). Однако, при сопоставимой стойкости, область практического использования этих групп стандартов существенно различается. Область использования криптоалгоритмов, установленных российскими стандартами фактически ограничена взаимодействием с российскими государственными организациями и мало используется в деловой практике. В тоже время криптоалгоритмы, установленные американскими стандартами, используются как на территории США, так и в международной деловой практике, в том числе и на территории России.

Такое положение дел во многом сложилось вследствие разных подходов к выбору стандартизируемых криптоалгоритмов.

Два аспекта открытости криптографического стандарта

Российский алгоритм шифрования был разработан в одной из структур КГБ СССР [9] и был открыт только в 1994 году. В стандарте не все характеристики алгоритма точно определены, что может в отдельных случаях привести к снижению криптостойкости при использовании «слабых» ключей и таблиц замены. Хотя, на данный момент, не известны эффективные методы криптоанализа при использовании «сильных» ключей и таблиц замены, изначальная закрытость и небольшая область применения привели к недостаточной изученности криптоалгоритма.

Американский алгоритм шифрования AES был выбран в результате открытого двухступенчатого международного конкурса, в ходе которого экспертное сообщество исследовало криптостойкость предложенных на конкурс алгоритмов и практические аспекты их использования.

Открытость и полнота описания криптографического стандарта важна не только для обеспечения единообразного выполнения криптографического алгоритма (что обеспечивается стандартом как таковым), но и для обретения уверенности в криптостойкости алгоритма, поскольку единственным способом убедиться в этом, является всестороннее экспертное изучение алгоритма [10].

Таким образом, полная открытость алгоритма шифрования AES, его всестороннее изучение множеством экспертов и последующие широкое применение этого алгоритма позволило убедиться в его практической стойкости. Такая уверенность международных экспертов в практической стойкости российского алгоритма шифрования ГОСТ-28147 отсутствует, хотя теоретическая стойкость обоих алгоритмов сопоставима.

Использование криптостандартов в протоколах сети Интернет

Следствием международного признания национальных криптостандартов США и уверенности в их стойкости стало использование данных алгоритмов в других международных стандартах и протоколах, в частности, в спецификациях и стандартах протоколов Интернет, которые разрабатывает *Internet Engineering Task Force (IETF)*. Рассмотрим некоторые из этих протоколов:

- Security Architecture for the Internet Protocol (IPsec) – набор протоколов, обеспечивающих надежную передачу данных по протоколу IP [11] – использует криптоалгоритмы AES, SHA (определен в стандарте SHS), DES.
- The Secure Shell (SSH) Protocol – обеспечивает безопасное удаленное управление и безопасную работу других сетевых служб в незащищенной сети [12] – использует криптоалгоритмы AES, DSS.

- The Transport Layer Security (TLS) Protocol – обеспечивает защищенную передачу данных в сети Интернет [13] – использует криптоалгоритмы AES, DSA (DSS), SHA (SHS), DES.
- S/MIME Message Specification (Secure/Multipurpose Internet Mail Extensions) – это спецификация, которая согласовывает способ передачи защищенных данных посредством электронной почты [14] – использует криптоалгоритмы SHA (SHS), DES.

Как видно из приведенного перечня, актуальные версии сетевых протоколов поддерживают только американские криптостандарты.

Влияние криптостандартов на конкурентоспособность криптографического обеспечения

Любое программно-аппаратное обеспечение, поддерживающее протоколы сети Интернет, обязано поддерживать и американские криптографические стандарты. Отечественное программно-аппаратное обеспечение с поддержкой российских криптоалгоритмов оказывается менее конкурентоспособным и практически не подлежит экспорту на мировой рынок. Таким образом, обладая сильной криптографической школой, Россия недостаточно эффективно использует этот интеллектуальный ресурс и ее влияние в этой области ограничено.

В последнее время предприняты попытки исправить сложившуюся ситуацию и добиться признания российских криптостандартов путем поддержки их международными стандартами, в частности, принят стандарт IETF RFC 4357, придающий российским стандартам статус международных для их возможного использования в протоколах Интернет [15, 16] и разработан предварительный стандарт, обеспечивающий поддержку российских криптоалгоритмов в протоколе TLS [16].

Заключение

Открытость криптографического стандарта важна для обеспечения совместимости криптографического обеспечения и коммуникаций и для обретения уверенности в криптостойкости алгоритма, установленного

стандартом. Большая уверенность международного экспертного сообщества в криптостойкости алгоритмов, установленных американскими стандартами, обеспечило их международное признание и использование в протоколах сети Интернет. Криптографическое обеспечение, реализующее совместимость на основе российских криптостандартов, имеет низкую конкурентоспособность на мировом рынке.

Литература:

1. Федеральный закон от 8 августа 2001 г. N 128-ФЗ "О лицензировании отдельных видов деятельности" (с изменениями и дополнениями) [Электронный ресурс]. – Режим доступа: <http://www.garant.ru/law/12023874-000.htm>.
2. FIPS 197. Advanced Encryption Standard [Электронный ресурс]. – Режим доступа: <http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf>.
3. FIPS 186-2. Digital Signature Standard (DSS) [Электронный ресурс]. – Режим доступа: <http://csrc.nist.gov/publications/fips/fips186-2/fips186-2-change1.pdf>.
4. FIPS 180-3. Secure Hash Standard (SHS) [Электронный ресурс]. – Режим доступа: http://csrc.nist.gov/publications/fips/fips180-3/fips180-3_final.pdf.
5. FIPS 46-3. Data Encryption Standard (DES) [Электронный ресурс]. – Режим доступа: <http://csrc.nist.gov/publications/fips/fips46-3/fips46-3.pdf>.
6. ГОСТ 28147-89. Системы обработки информации. Защита криптографическая. Алгоритм криптографического преобразования. – М.: Госстандарт СССР, 1989.
7. ГОСТ Р 34.11-94. Информационная технология. Криптографическая защита информации. Функция хэширования. – М.: Издательство стандартов, 1994.
8. ГОСТ Р 34.10-2001. Информационная технология. Криптографическая защита информации. Процессы формирования и

проверки электронной цифровой подписи. – Режим доступа: <http://protect.gost.ru/document.aspx?control=7&id=131131>.

9. Винокуров, А. Алгоритм шифрования ГОСТ 28147-89, его использование и реализация для компьютеров платформы Intel x86 [Электронный ресурс] / А. Винокуров. – Режим доступа: http://www.enlight.ru/crypto/download/articles/vinokurov/28147_89.zip.

10. Schneier B. Open Source and Security [Электронный ресурс]. – Режим доступа: <http://www.schneier.com/crypto-gram-9909.html#OpenSourceandSecurity>.

11. Security Architecture for the Internet Protocol (IPsec) [Электронный ресурс]. – Режим доступа: <http://www.ietf.org/rfc/rfc4301.txt>.

12. The Secure Shell (SSH) Protocol [Электронный ресурс]. – Режим доступа: <http://www.ietf.org/rfc/rfc4251.txt>.

13. The Transport Layer Security (TLS) Protocol [Электронный ресурс]. – Режим доступа: <http://www.ietf.org/rfc/rfc5246.txt>.

14. S/MIME Message Specification [Электронный ресурс]. – Режим доступа: <http://www.ietf.org/rfc/rfc2311.txt>.

15. Additional Cryptographic Algorithms for Use with GOST 28147-89, GOST R 34.10-94, GOST R 34.10-2001, and GOST R 34.11-94 Algorithms [Электронный ресурс]. – Режим доступа: <http://www.ietf.org/rfc/rfc4357.txt>.

16. Поддержка российских криптоалгоритмов в OpenSSL [Электронный ресурс]. – Режим доступа: http://www.cryptocom.ru/OpenSource/OpenSSL_rus.html.

Раздел II. E-learning: технологии, стандарты, перспективы

Ю.С. Лактионова
ГОУ ВПО «Магнитогорский Государственный университет»
г.Магнитогорск

ОСНОВНЫЕ АСПЕКТЫ И ВОЗМОЖНОСТИ

ТЕОРИИ РАЗВИВАЮЩЕГО ОБУЧЕНИЯ И ТЕХНОЛОГИИ E-LEARNING ДЛЯ ПОВЫШЕНИЯ ЭФФЕКТИВНОСТИ ОБУЧЕНИЯ ИНФОРМАТИКЕ

Быстрый количественный и качественный рост информационно-технологического пространства объективно приводит к тому, что проблема информатизации всех сфер жизнедеятельности общества постепенно становится одной из наиболее важных. По мере возрастания объема информации, развития компьютерной техники и новых информационных технологий человечество не просто осознает фундаментальную роль информации в общественном развитии, но и проводит активные научные разработки тех направлений в системе «человек-компьютер», которые связаны с мышлением, познавательной самостоятельностью, творческой активностью при использовании средств компьютерных технологий, ибо информатизация накладывает свой отпечаток не только на организацию знания в современной картине мира, но и на способы и приемы мышления, организацию самостоятельной деятельности. Это в свою очередь находит отражение в системе образования, требуя кардинальных изменений в формировании познавательной самостоятельности, творческой активности, критического и рефлексивного мышления учащихся.

Возникает необходимость поиска таких педагогических технологий, которые могли бы дать возможность учащемуся ориентироваться и действовать в современных условиях, которые требуют от человека усилий и способности к саморазвитию, создания более эффективных форм учебной деятельности, которые дадут предпосылки для организации полноценных процессов самообразования. Мы считаем, что теория развивающего обучения,

в основу которой положен принцип деятельностного подхода, является для этого вполне состоятельной.

В связи с этим в основу образовательной программы школы по информатике, наряду с другими дисциплинами, может быть положена система развивающего обучения (РО). РО - это целостная психолого-педагогическая система, альтернативная традиционной школе. Теория развивающего образования оперирует такими фундаментальными понятиями как понятие деятельности, идеального и сознания, личности; дает логико-психологические основания эмпирического и теоретического мышления; описывает учебную деятельность школьников, осуществление которой способствует их психическому развитию.

Современные стратегические цели образования акцентируют внимание на формировании творческой, самостоятельной личности, развитии ее как активного субъекта собственной жизни и деятельности, свободно проявляющей себя в духовной и социальной сферах, и содержат указание на необходимость изменений знаниево-функциональной направленности образования на личностно-развивающую. В настоящее время в российской начальной школе существует три государственных образовательные системы (традиционная, система Л.В.Занкова, система Д.Б.Эльконина-В.В.Давыдова). Практически все авторские коллективы претендуют на то, чтобы их труды могли называться личностно-развивающими программами. Этот статус достигается в основном за счет изменения форм и методов обучения. Подходы к построению содержания образования практически в своих основаниях не изменились. Образовательная же система Д.Б.Эльконина - В.В.Давыдова является одной из немногих образовательных систем, которая ставит своей основной стратегической целью интеллектуальное развитие учащихся и способная решить задачи, поставленные в Концепции модернизации российского образования за счет, прежде всего, изменения содержания образования.

Школа по своей функции нацелена на развитие общества, она должна обеспечивать такое развитие, в котором предоставляются широкие возможности доступа к данным и фактам, образование должно готовить субъекта к активному использованию этой информации, то есть вооружить его возможностями ее сбора, отбора, упорядочивания, управления и использования, подготовить учащихся к профессиональной деятельности в информационном обществе. Что невозможно осуществить без изучения курса информатики.

Наиболее продуктивной, на наш взгляд, может стать методология построения дидактической системы обучения информатике, базирующаяся на комплексном, сбалансированном учете основных факторов, определяющих содержание образовательной области, на комплексном анализе структуры изучаемой области действительности и структуры деятельности человека в данной области с учетом принципов развивающего обучения. Причем, деятельностный подход и принципы РО целесообразно использовать на всех этапах построения дидактической системы: при определении целей обучения, отборе содержания, определении основных принципов построения методики обучения, выборе форм представления материала, построении системы учебных задач, выборе форм и средств обучения, организации контроля результатов обучения.

Последовательное применение данной методологии позволит перейти от обучения информатике, носящего преимущественно информационный характер и направленного в основном на исполнительскую деятельность, на формирование личности, умеющей ориентироваться и принимать обоснованные решения в условиях современной информационной среды, владеющей приемами творческой деятельности и способной не только усваивать готовое знание, но и генерировать новое.

Такой подход позволит решить и другую достаточно острую в настоящее время проблему, а именно, в полной мере обосновать обобщенную структуру деятельности в области информатики в отличие от

той ее части, которая составляет профессиональное образование, связанное в большей степени с computer science.

Применение методологии деятельностного подхода при построении дидактической системы с необходимостью потребует конкретизации основных положений, в соответствии с которыми выбираются методы обучения. Одно из направлений связано с тем, что необходимо определить принципы разбиения содержания на "порции учебного материала". В частности, каждая порция, подлежащая усвоению, должна сочетать в себе как "знаниевые", так и "деятельностные" аспекты.

Полагаем, что процесс обучения информатике с использованием дидактического обеспечения школьного курса информатики на основе теории развивающего обучения будет более эффективным при наличии определенного комплекса педагогических условий. Одним из которых является обеспечение системы электронной поддержки деятельности учащихся и учителей при работе с дидактическим комплексом.

Тенденцией современного этапа информатизации образования является всеобщее стремление к использованию в процессе обучения средств информационных и коммуникационных технологий (ИКТ), таких как электронные справочники, энциклопедии, обучающие программы, средств автоматизированного контроля знаний обучаемых, компьютерных учебников и тренажеров. И это вполне обосновано. Использование ИКТ в системе общего среднего образования приводит к повышению эффективности обучения за счет изменения уровня его индивидуализации и дифференциации, использования дополнительных мотивационных рычагов. Во многих случаях использование современных средств ИКТ дает возможность дифференциации процесса обучения школьников за счет использования средств и технологий выбора заданий разного уровня, организации самостоятельного продвижения по темам курса успевающим школьникам и возврату к недостаточно изученному материалу отстающим ученикам. Учитывая возможные факторы индивидуализации и

дифференциации обучения, средства ИКТ, обоснованно примененные в системе общего среднего образования, способствуют организации личностно – ориентированного обучения.

Средства информационных и коммуникационных технологий все чаще применяют в школах для автоматизации процессов контроля и измерения результативности обучения школьников. Современные измерительные материалы и средства измерений все чаще называют тестами. Основными преимуществами заданий, представляемых в тестовой форме, по сравнению с традиционными задачами и вопросами, являются краткость, логическая структурированность, стандартизованность и единая относительно простая процедура проведения тестирования и оценки его результатов. Именно эти преимущества делают тесты наиболее пригодными для оценки результатов обучения школьников и проверки соответствия этих результатов требованиям государственных стандартов образования. Подобные формы представления тестовых заданий максимально упростили их формализацию и позволили создать средства ИКТ, автоматизирующие процессы педагогического контроля и измерений.

Более того, использование ИКТ в процессе обучения делает более доступными образовательные услуги, в особенности для детей, не имеющих возможности по различным причинам учиться в школе в традиционном режиме.

E-learning – это использование современных информационных технологий для эффективной доставки знаний и развития человеческих ресурсов. Сегодня через использование доступных технологических решений, интернет, мобильных телефонов, а также новых программных средств возможно донесение знаний и до географически сильно отдаленных субъектов, причем делать это быстро, наглядно и эффективно.

Исходя из всего вышесказанного, для повышения эффективности обучения информатике в школе было решено использовать не только

технологии развивающего обучения, но и возможности технологии e-learning.

Использование последнего предусматривает разработку и информационное наполнение электронного учебно-методического комплекса (ЭУМК) по информатике для 11 классов базового уровня среднего (полного) общего образования. ЭУМК будет включать следующие элементы:

- Рабочая программа курса.
- Тематический план.
- Учебное пособие и методические рекомендации по изучению дисциплины.
- Система текущего и итогового контроля.

Успешное изучение информатики, как и любого другого предмета, зависит не столько от объема информации, сколько от ее структурирования и логики подачи, методов усвоения и видов деятельности учащихся, таким образом полагаем, что использование ЭУМК в процессе обучения позволит не только смягчить проблему недостаточности и ограниченности информации, необходимой для формирования требуемых знаний, умений, навыков, если таковая имеется, но и поможет структурировать уже имеющиеся знания в изучаемой области, т.о. позволит будущим выпускникам успешно подготовиться к сдаче ЕГЭ по информатике. Использование ЭУМК, а точнее его системы текущего контроля в организации учебного процесса позволит оперативно реагировать на успеваемость учащихся и устранять «пробелы». Наличие рейтинговой системы оценки окажет положительное влияние на стимулировании активности учащихся, дополнительной заинтересованности их в результатах деятельности.

Литература:

1. Беликов, В.А. Философия образования личности: Деятельностный аспект: монография/ Беликов В.А. – М.: Владос, 2004. – 357 с.

2. Давыдов, В.В. Теория развивающего обучения./В.В. Давыдов. – М., 2002.
3. Воронцов, А.Б., Чудинова, Е.В. Учебная деятельность. Введение в систему Д.Б. Эльконина - В.В. Давыдова./А.Б. Воронцов, Е.В. Чудинова. – М., 2004.
4. Башмаков, А.И., Башмаков, И.А. Разработка компьютерных учебников и обучающих систем./А.И. Башмаков, И.А. Башмаков. – М.: Информационно-издательский дом «Филинь», 2003. – 616 с.

В.Б. Лапшина, В.Н. Макашова
ГОУ ВПО «Магнитогорский Государственный университет»
г.Магнитогорск

РАЗВИТИЕ РЫНКА E-LEARNING УСЛУГ В РОССИИ

Развитие технологий оказывает непосредственное влияние на все сферы нашей жизни. Образование является одним из тех ярких примеров. Актуальной составляющей современной образовательной системы, в данный период времени становится электронное обучение, обеспечивая принципиально новые возможности в доступе к образовательным информационным ресурсам, в организации управления образовательными процессами, в актуализации образовательных ресурсов и управления ими, в организации новых форм образования с использованием дистанционных форм обучения, значительно расширяя возможности традиционной образовательной системы.

Электронное обучение (e-learning) – это перспективная модель обучения, основанная на использовании новых мультимедийных технологий и Интернета для повышения качества обучения путем облегчения доступа к ресурсам и услугам, а также обмена ими и совместной работы на расстоянии.

В настоящее время интерес к электронному обучению неуклонно возрастает, электронное обучение стало активно использоваться не только в образовательном процессе учебных заведений, но и в повышении

квалификации персонала организации и предприятия. E-learning рассматривается как основное средство обучения на протяжении жизни, которое является неотъемлемой составной частью построения и функционирования общества.

В России рынок e-learning услуг (электронное обучение, дистанционное обучение) начал развиваться с начала 2000-х годов (1). Одним из лидирующих университетов в области электронного обучения стал Московский государственный университет экономики статистики и информатики (МЭСИ). В корпоративном секторе дистанционные электронные курсы первыми стали внедрять такие компании как, Газпром, Федеральное казначейство России, FORD.

В 2005 году в России успешно завершились испытания отечественной системы дистанционного обучения СДТ REDCLASS на соответствие международному стандарту SCORM 1.2. Данный стандарт является признанным во всем мире стандартом в сфере e-learning и поддерживается практически всеми ведущими производителями систем дистанционного обучения, а СДТ REDCLASS стала первой и пока единственной из сертифицированных на международном уровне российских систем e-learning (2).

Однако по сравнению с США и Европой корпоративное электронное обучение в нашей стране находится в зачаточном состоянии: в рейтинге 2006 года в области e-learning Россия занимала 52-е место в мире из представленных 68 стран. По итогам 2008 года в США объем рынка электронного обучения достиг примерно 30 млрд. долларов, тогда как в России объем рынка оценивался по разным подсчетам порядком 100 млн. долларов (3).

Сейчас, по словам президента международного консорциума «Электронный университет» В. Тихомирова, Россия занимает в рейтингах крайне низкие позиции по уровню электронного обучения и полностью выпала из процессов создания национальных и международных

образовательных консорциумов, распространяющих опыт организации новых форм учебного процесса с использованием ИТ. Одной из главных причин сложившейся ситуации, считает господин Тихомиров, является отсутствие нормального правового поля для инициатив по организации обучения в электронной среде, что в последнее время отвратило от подобных попыток даже самые смелые и активные вузы (4).

Директор по маркетингу Компании ГиперМетод Д. Кречман называет в качестве одной из причин, по которой внедрение e-learning в России не получает пока должного признания – отсутствие качественного готового контента. Наличие электронных курсов, тестов, тренингов высокого качества могло бы значительно увеличить число пользователей e-learning (5).

Однако некоторый оптимизм в отношении ситуации с электронным обучением в нашей стране вселяет внимание президента РФ Дмитрия Медведева к использованию ресурсов сети Интернет и ИТ в образовании. Недавно при Министерстве образования РФ была создана рабочая группа по подготовке нормативных документов в области электронного обучения, появление которых может способствовать более активному проникновению e-learning в российские учебные заведения.

В настоящее время возможность получения образования дистанционным путем предоставляют около 40% вузов. Наиболее крупные по количеству студентов из них: Центр дистанционного обучения Академии Народного Хозяйства при Правительстве РФ, Moscow Business School, Международный институт менеджмента ЛИНК, Институт магистерской подготовки МЭСИ. В образовательной сфере e-learning продолжит дополнять традиционный очный вариант обучения, и в большинстве случаев наиболее целесообразным останется смешанное обучение, когда одни курсы, в зависимости от их специфики, изучаются традиционным образом, а другие – дистанционно.

На данный момент дистанционное обучение персонала с успехом реализовано на таких крупных предприятиях, как «Российские железные

дороги», «СеверСталь», «Норильский никель», «РусАл», «ВымпелКом», «УралСиб», «Связьинвест» и др. Возможности дистанционного обучения были задействованы при переобучении сотрудников Государственной Думы Федерального собрания РФ, Центрального банка России, Внешторгбанка и ряда других организаций (4). Следует отметить, что на сегодняшний момент в корпоративном секторе рынка e-learning наибольшим спросом пользуется краткосрочное обучение и особенно популярны готовые электронные курсы по IT-технологиям и продажам.

Одним из ведущих мероприятий в области электронного обучения в России является ежегодная выставка и конференция eLearnExpo. Цель данного мероприятия познакомить участников с самыми современными технологиями обучения с использованием информационно-коммуникационных технологий. Экспонентами выставки являются компании, производящие программные продукты для e-learning, компании, специализирующиеся на подготовке образовательного контента, провайдеры проектных и консалтинговых услуг в сфере e-learning, поставщики аппаратных платформ для различных моделей и решений в сфере e-learning (6).

Согласно прогнозам экспертов и заказчиков курсов электронного обучения, в 2009 году рынок электронного обучения в России вырастет на 20% на фоне сокращения объема аудиторного обучения в компаниях (тренингов и семинаров) на те же 20% (7).

Наиболее перспективными в плане внедрения e-learning в России эксперты и аналитики считают корпоративный сектор, государственные структуры и центры переподготовки кадров. Для получения базового образования в российских вузах вариант e-learning, видимо, пока что малоперспективен – прежде всего из-за значительного снижения числа абитуриентов.

В 2010 году, если верить специалистам из American Educational Research Association, две трети всего обучения в мире будет осуществляться

дистанционно. Скорее всего, данный прогноз стоит считать слишком оптимистичным, но несомненно одно – электронное обучение стало достойной альтернативой традиционному и в отдельных сферах, прежде всего в корпоративной и государственной, ему будут отдавать явное предпочтение, поскольку это единственный способ быстрого обучения при минимальных затратах (2).

Литература

1. Портал BISHHELP. /Дистанционное обучение и e-learning в России. [Электронный ресурс]. URL: <http://www.bishelp.ru/uprbiz/obuchenie/dist/index.php>
2. Шляхтина, С. Перспективы развития дистанционного обучения в мире и в России. / С. Шляхтина [Электронный ресурс]. <http://www.compress.ru>
3. Ходонова, А. На электронное обучение перейдут 20% учащихся. / А. Ходонова [Электронный ресурс]. URL: <http://bfm.ru/articles/2009/08/23/na-elektronnoe-obuchenie-perejdut-20-uchashhihsja.html>
4. Портал Trainings.ru. /Научный руководитель МЭСИ: Россия занимает в рейтингах крайне низкие позиции по уровню электронного обучения. [Электронный ресурс]. URL: <http://www.trainings.ru/library/articles/?id=11949>
5. Портал Trainings.ru. /Прошлое, настоящее и будущее e-learning в России. [Электронный ресурс]. URL: <http://training.bl.by/articles/article81645/>
6. E-learnExpro Moscow. <http://www.elearnexpo.ru>
7. Смирнова, Е. Держи дистанцию / Е. Смирнова [Электронный ресурс]. URL: <http://www.archive-online.ru/read/hrdirector/80>

АСПЕКТЫ ИНТЕГРАЦИИ ТРАДИЦИОННОЙ И ДИСТАНЦИОННОЙ ТЕХНОЛОГИЙ ОБУЧЕНИЯ В ВУЗЕ

Концепция модернизации российского образования реализуется сегодня посредством исполнения целого ряда программ, видное место среди которых занимают программы, направленные на информатизацию образования. Реализация этих программ привела к коренным изменениям в оснащенности образовательных учреждений средствами информационных и коммуникационных технологий, расширяется тенденция интеграции этих средств в учебный процесс вузов, происходит накопление опыта создания цифровых образовательных ресурсов. Возникла и развивается Интернет-поддержка очного учебного процесса, все большее распространение получает технология дистанционного обучения, при которой обучающийся и преподаватель пространственно отделены друг от друга, находясь при этом в постоянном взаимодействии. Ведется интенсивная работа в направлении развития теории и методики дистанционного обучения, которые базируются на основных положениях традиционного образовательного процесса и расширяют его возможности.

В 2005 году приказом Министерства образования и науки Российской Федерации «Об использовании дистанционных образовательных технологий» был утвержден Порядок использования дистанционных образовательных технологий. При этом, использование технологии дистанционного обучения (ТДО) не исключает возможности проведения учебных, лабораторных и практических занятий, практик, текущего контроля, промежуточной и итоговой аттестаций путем непосредственного взаимодействия педагогического работника с обучающимся, что является, на наш взгляд, одним из подходов в реализации технологий смешанного обучения. Соотношение объема проведенных учебных, лабораторных и

практических занятий с использованием ТДО или путем непосредственного взаимодействия педагогического работника с обучающимся определяется образовательным учреждением.

Стирание грани в современном образовании между технологически продвинутым традиционным и дистанционным обучением возможно на основе использования международного опыта, а именно - Learning Technology Standards.

- IEE Learning Technology Standards Committee.
- ADL - Advanced Distributed Learning - проект Министерства Обороны США в области дистанционного обучения.
- AICC - Aviation Industry CBT (Computer Based Training) Committee - международная организация специалистов в области технологии обучения. Результатом работы являются технические руководства (AGR AICC Guidelines & Recommendations), а так же более детальные технические отчеты и другая аналитика.
- IMS Global Learning Consortium - представляет собой некоммерческий консорциум организаций разработчиков e-learning решений с целью создания открытых спецификаций. Данные спецификации определяют различные компоненты систем электронного обучения и базируются на XML. Часть спецификаций стали стандартами де-факто и легли в основу других разработок.
- LRN - является первым коммерческим применением спецификации IMS Content Packaging Specification. LRN была разработана и поддерживается компанией Microsoft и в настоящее время не получила широкого распространения.
- SCORM - Sharable Content Object Reference Model - является инициативой ассоциации ADL. Обучающие объекты совместимые со SCORM могут быть повторно использованы в различных системах обучения. SCORM состоит из нескольких разделов или книг,

описывающих различные аспекты (Run Time Enviroment, Content Aggregation Model, Sequencing and Navigation).

Существуют разные модели реализации технологии дистанционного обучения. В последние годы все большее распространение получают четыре модели, основанные на видеосвязи, спутниковой связи, компьютерных и коммуникационных технологиях (региональных, глобальных), с различными дидактическими возможностями в зависимости от используемых конфигураций (с использованием, в основном, текстовых файлов, мультимедийных технологий, видеоконференций); компьютерных и коммуникационных технологий в сочетании с кейс-технологиями: интеграция очных и дистанционных форм обучения; сетевое обучение (Интернет-технологии), которое в свою очередь может включать: автономные курсы, информационно-образовательные среды (виртуальные кафедры, школы, университеты); сетевое обучение и кейс-технологии; интерактивное видео (спутниковая связь, интерактивное телевидение). Все указанные модели могут решать конкретные дидактические задачи в определенных условиях.

В процессе подготовки ИТ-специалиста традиционные формы обучения могут быть дополнены средствами ТДО. Углубленное изучение отдельных вопросов теории с использованием традиционно принятых форм теоретических семинаров разных уровней, конференций, симпозиумов и т.д. дополняется участием в телеконференциях, проводимых с помощью коммуникационных сетей, в том числе и посредством мировой сети Internet. Тематические конференции, проводимые в рамках глобальной компьютерной сети, несоизмеримо увеличивают возможности общения специалистов единого профиля и позволяют оперативно получать новейшую информацию в короткие сроки. То же самое относится и к практическим занятиям с применением ТДО. Обучение с целью профессионального совершенствования можно эффективно организовать на основе применения интеллектуальных тестирующих программ, которые позволяют полностью

сформировать реакцию на поставленный вопрос на требуемом интеллектуальном уровне.

Таким образом, сопоставительный анализ традиционных технологий с ТДО позволяет заключить, что последние содействуют: более полному раскрытию интеллектуальных способностей личности; индивидуализации процесса обучения; диагностике и самодиагностике профессиональных качеств на основе использования тестирующих программ; формированию и использованию в процессе обучения виртуального социального и профессионального пространства; мониторингу профессионального состояния с использованием локальных и глобальных сетей, включая мировую информационную сеть Internet; коммуникативности в образовательной информационной среде на микро и макроуровнях, включая телеконференции и публикации в сети Internet.

Итак, учитывая положительного влияния ТДО на процесс подготовки ИТ-специалиста, и в соответствии с концепцией модернизации образования, наиболее перспективным нам видится сочетание в образовательном процессе традиционных педагогических и дистанционной технологий обучения на основе взаимообогащения их возможностей. Реализация данной модели позволит разгрузить аудиторные занятия от рутинных видов деятельности, требующих разного времени для разных учащихся (формирование навыков), позволит более тщательно и самостоятельно подготовиться к обсуждению важных вопросов по предмету изучения, восприятию лекционного материала, к дискуссиям, осуществить поиск и анализ дополнительной информации для проектной, исследовательской деятельности. Аудиторные же занятия в большей степени посвятить обсуждению изучаемого материала, дискуссиям по проблемам, связанным с данной темой, лабораторным, практическим работам, защите проектов, результатов исследовательской деятельности.

Возможности модели интегрированного обучения достаточно перспективны, хотя и требуют определенных организационных и административных решений.

Наиболее перспективными моделями обучения с использованием технологии дистанционного обучения (на основе анализа международного и отечественного опыта) на наш взгляд являются: модель сетевого дистанционного обучения; модель смешанного обучения (blended learning). Для них разработаны описания на основе модели 3-го уровня LTSA (стандарт IEEE P1484.1).

В модели сетевого дистанционного обучения (по LTSA) центральным компонентом является среда дистанционного обучения, основным каналом доставки образовательного контента - Интернет (Интранет), предусмотрены учет объема мультимедийного контента, поддержка работы преподавателей, программистов и дизайнеров по подбору и формированию образовательного контента, взаимодействия обучаемого с преподавателем в on-line и off-line режимах (e-mail, форумы, чаты), видеоконференции, оценивание результатов обучения с помощью Интернет-тестирования, фиксации промежуточных результатов работы учащихся с ресурсами, взаимодействия с тьютором, учет результатов образовательной деятельности в обязательном порядке ведется в электронной форме.

В модели смешанного обучения (по LTSA) многоканальность доставки образовательного контента учащимся осуществляется с помощью используемых ИКТ технологий: электронные образовательные ресурсы (ЭОР) на локальных носителях (CD, DVD и пр.); в сети (локальной или Интернет); на аудиторных занятиях используются презентации, фрагменты ЭОР, анимация, виртуальные практикумы и пр.; ведется оперативное тиражирование печатных материалов, поддержка методической работы преподавателя при обучении в ИКТ насыщенной среде (электронная библиотека, медиатека, электронный каталог), имеется расширенный набор средств взаимодействия обучаемого с преподавателем: традиционное

общение в аудитории, электронная почта, образовательные Интернет-форумы, трансляция лекций, видеоконференции и пр.; оценивание результатов обучения – традиционное в сочетании с характерным для дистанционного обучения (тестирование, учет форумов и пр.). Цели обучения при смешанной форме остаются прежними, изменяются средства и методы достижения целей. Система контроля знаний получает новые возможности: использование контролирующих систем в сочетании с традиционными методами.

Таким образом, смешанная модель обучения – это модель использования распределенных информационно-образовательных ресурсов в очном обучении с применением элементов технологий асинхронного и синхронного дистанционного обучения. Практикуется в качестве элемента очного обучения при проведении аудиторных занятий и в самостоятельной работе студентов. То есть смешанное обучение наследует достоинства дистанционного обучения и исключает его недостатки. Смешанное обучение использует самые разнообразные технологии и методы как традиционные, так и интерактивные: лекционные лаборатории, компьютерные презентации, компьютерное обучение и обучение через Интернет. Эти методы используются как по отдельности, так и в сочетании друг с другом. В результате вышеизложенного можно отметить следующие преимущества смешанного обучения: двустороннее общение, учащийся активен, деятелен, вовлечен в учебный процесс, учащийся знаком со структурой, учащийся находится под чутким руководством, текст учебного пособия написан в дружеской и ободряющей манере, учащийся применяет приобретенные знания и умения, задания и упражнения рассеяны по всему тексту, текст поделен на небольшие разделы, контрольные задания предусмотрены, учащийся получает отзывы о своих успехах. Смешанное обучение предполагает сохранение общих принципов построения традиционного учебного процесса. Соотношение долей использования традиционных и дистанционных технологий обучения определяется готовностью к

подобному построению учебного процесса образовательного учреждения в целом, а также желанием и техническими возможностями студентов. Такой подход носит название «гибкое обучение» (flexible learning).

Очевидно, что, внедряя в учебный процесс современные методы обучения можно значительно повысить его качество, сделать учебный процесс более гибким, стимулировать студентов к самостоятельной работе.

А.В. Меркулова
ГОУ ВПО «Магнитогорский государственный университет»
г.Магнитогорск

E-LEARNING: СУЩНОСТЬ ПРОБЛЕМЫ ВНЕДРЕНИЯ

Введение

Сейчас во всем мире повсеместно происходит информатизация общества, т.е. глобальное внедрение информационных технологий. Одним из важнейших шагов информатизации общества является информатизация образования, поскольку именно в сфере образования подготавливаются специалисты, которые не только формируют новую информационную среду общества, но и которым предстоит жить и работать в этой новой среде.

В последние несколько лет одним из приоритетных направлений развития современного российского образования является внедрение и повсеместное использование средств электронного обучения (e-learning).

Во всем мире система электронного обучения очень бурно развивается, а у нас в России, к сожалению, многие о ней не осведомлены должным образом. Пионером e-learning в России стал Московский государственный университет экономики, статистики и информатики, в котором E-learning-технологии работают уже более 12 лет. Сейчас это направление начинают развивать многие другие вузы.

В Магнитогорском госуниверситете в марте 2009 г. состоялся научно-практический семинар, посвященный столь актуальному направлению. Организатором мероприятия выступала лаборатория открытых систем

факультета информатики при поддержке Секции открытых систем Совета РАН «Высокопроизводительные системы, научные телекоммуникации и информационная инфраструктура». На семинаре были представлены доклады, которые рассматривали e-learning с разных точек зрения: науки, педагогики и технологий реализации. Также своим опытом поделились докладчики, занимающиеся подготовкой кадров для ОАО «ММК»: КЦПК «Персонал», и ООО «Корпоративные системы».

Рассмотрим основные проблемы, обсуждаемые в рамках семинара «E-learning: технологии, стандарты, перспективы».

1. Проблема определения понятие e-learning.

Электронное обучение (ЭО), иначе, e-learning, зародившись в конце 20-го столетия, вошло в 21 век как одно из наиболее эффективных и перспективных систем подготовки специалистов. Появление и активное распространение дистанционных форм обучения является адекватным откликом систем образования многих стран на происходящие в мире процессы интеграции, движение к информационному обществу.

Что же такое e-learning? По определению специалистов ЮНЕСКО: «Это обучение с помощью Интернета и средств мультимедиа». Оно зарождалось как система поддержки дистанционного образования. Интернет тогда являлся единственным коммуникативным средством между студентом и вузом. Поэтому многие до сих пор путают его с дистанционным обучением. Но сегодня технологии ушли далеко вперед. В настоящее время e-learning — это учебный материал и обязательное общение студента и преподавателя посредством Интернета, которое может проходить как в виде видеоконференции в режиме on-line, так и в интерактивной форме. E-learning — смешанное обучение, объединение очного и виртуального сценариев, ставшее основной формой взаимодействия преподавателя и учащегося. В данном случае присутствует электронная компонента образования, то есть компьютер, Интернет, информационные технологии, а также то, что непосредственно поддерживается общением человека с человеком.

2. Проблемы внедрения e-learning в России.

В настоящее время созданы технические предпосылки для широкого использования электронного обучения (ЭО) в России.

Несмотря на это, в сфере ЭО существует ряд проблем. Первая проблема, от которой, собственно, тянутся и все остальные – это информированность: на данный момент в России очень мало граждан знают о существовании электронного образования, не говоря уже о его эффективности, следовательно, на e-learning как на услугу не существует должного спроса.

Другая, не менее важная проблема, - это техническое оснащение. Все категории потребителей, нуждающиеся в ЭО, по большей части имеют меньшую возможность доступа к сети, чем граждане, в ЭО, в принципе, не нуждающиеся.

Отсутствие отечественных стандартов и слабая адаптация зарубежных – основная причина создания неконкурентных образовательных систем и курсов.

Отсутствие кадров, т.е. очень мало специалистов (особенно в регионах), которые действительно понимают, что нужно делать при разработке электронного курса и как нужно изменять материалы традиционного.

На данный момент разработкой электронных курсов в России занимаются такие компании, как «Новый диск», «CSSe Train», «Websoft», но их целевая аудитория – это, прежде всего, работники крупных предприятий, а заказчики – такие гиганты, как «Русский алюминий», «Северсталь» и «Газпром».

Еще одна проблема – это оценка качества электронного курса. Существует список компонент, которые должен включать любой качественный e-learning контент (т.е. содержание этого курса) – это: руководство по изучению дисциплины; календарный план обучения; учебное пособие; тесты и задания для самопроверки; презентации; список

обязательных и дополнительных источников; вопросы или планы для форумов и чата; приемлемый дизайн и дружелюбный интерфейс. Однако конкретных критериев оценки каждой компоненты контента не существует.

В последнее время остро стоит проблема нормативно-правового регулирования. На данный момент в России нет законодательной базы, четко регулирующей отношения в сфере открытого образования. И, прежде всего, здесь существует проблема защиты интеллектуальной собственности.

Важно учитывать и человеческий фактор. Для получения ЭО необходима сильнейшая самодисциплина, ведь «ученик» самостоятельно изучает учебный курс, и именно от него зависит, закончит он его или нет. И, если для человека, не имеющего возможность получать образование в традиционной форме, такая проблема не стоит, то для среднестатистического абитуриента после школьных рамок и ограничений (в хорошем смысле этих слов) будет весьма сложно самоорганизоваться и пройти электронный курс самостоятельно.

3. Проблема рисков e-learning.

Электронное обучение имеет еще и некоторые специфические риски, о которых было достаточно полно изложено на семинаре:

- невозможность осуществления непосредственного контроля со стороны преподавателя за выполнением студентом учебных заданий;
- невозможность контроля качества образования со стороны обучаемого;
- риск мошенничества со стороны образовательного учреждения, предоставляющего возможности дистанционного образования;
- проблема защиты интеллектуальной собственности;
- проблема защиты информации, передаваемой в процессе обучения;
- длительное использование ИКТ в процессе обучения, потенциально провоцирующее возникновение различных негативных явлений в физическом, нравственном и духовном здоровье обучаемого.

4. E-learning и проблема интероперабельности.

В последние годы все большее значение во всем мире придается обеспечению интероперабельности.

Существует много определений интероперабельности, большинство из которых вводятся различными организациями, исходя из собственных интересов. Представляется целесообразным использовать определение, которое сформулировано в международном стандарте (ISO/IEC 24765:2008 Systems and Software Engineering Vocabulary), поскольку оно прошло процесс согласования между всеми заинтересованными сторонами. Это определение следующее: «Интероперабельность - способность двух или более систем или компонентов обмениваться информацией и использовать эту информацию».

Следует отметить, что проблемы обеспечения интероперабельности актуальны для систем всех масштабов – от GRID- систем до наносистем и для систем всех назначений, а именно систем, применяемых в науке, здравоохранении, образовании, государственном управлении, бизнесе, военном деле и др.

Внедрение технологий e-learning создает новую инновационную образовательную среду, для которой остро стоит проблема интероперабельности как традиционной системы обучения с электронной, так и систем электронного обучения между собой. Данная проблема порождает и определенный скепсис относительно перспектив развития e-learning и ряд других проблем, касающихся внедрения технологий e-learning, в частности, это проблема качества образования в новых условиях, проблема, связанная с экономической эффективностью и рисками внедрения данных технологий.

Заключение.

При подведении итогов, участники семинара отметили, что перспективы e-learning - в индивидуализированном подходе к обучению, в возможности использовать в обучении все богатство интеллектуальных ресурсов Сети, в общении с ведущими экспертами по всему миру, в

кооперации и, наконец, в максимальном приближении процесса повышения квалификации к рабочему месту специалиста. Сегодня существует огромное количество технологических решений для обучения, однако сами по себе они никогда работать не будут. Внедрение e-learning требует формирования стратегии, некоей последовательности шагов, которые позволят получить желаемый результат.

К.А. Рубан
ГОУ ВПО «Магнитогорский государственный университет»
г.Магнитогорск

ИНТЕРОПЕРАБЕЛЬНОСТЬ КОМПОНЕНТОВ ЭЛЕКТРОННОГО ОБУЧЕНИЯ

Европейская комиссия определяет электронное обучение (eLearning) как «использование новых технологий мультимедиа и Internet для повышения качества обучения за счет улучшения доступа к ресурсам и сервисам, а также удаленного обмена знаниями и совместной работы». Проще говоря, под eLearning подразумевается учебный процесс, в котором используются интерактивные электронные средства доставки информации, включая компакт-диски, корпоративные сети и Internet.

Для того, что бы оценить и понять, что же сегодня скрывается под понятием электронного обучения, необходимо заглянуть в историю его становления.

Истоки современного электронного обучения можно найти в появившихся в начале 90-х технологиях компьютерных тренингов (computer-based training, СВТ). Курсы на CD-ROM предназначались для автономного использования или, в редких случаях, в локальной сети клиент-серверной архитектуры.

Развитие гипертекстовых и медиа-технологий привело к созданию учебно-методических комплексов, которые могли использоваться как

локально, на компьютере обучаемого, так и удаленно, используя все плюсы глобальных сетей и клиент-серверной архитектуры. Именно с этого момента возникла необходимость в организации системы для создания, хранения и передачи учебных материалов, данных по пользователям, результатов обучения и анализа всей получаемой информации.

Так в конце 90-х возникли системы категории Learning Management System (LMS), включающие средства не только для организации и контроля использования учебных материалов, но и для администрирования учебного процесса в целом, в том числе, его традиционных форм. Стремление к тому, чтобы данные разных производителей, в том числе и курсы, разработанные заказчиками самостоятельно, были совместимы с разными LMS-платформами, стимулировало работы по стандартизации.

Три поколения СЭО.

Поколение 1. Системы электронного обучения первого поколения, стали появляться примерно в 1993 году и, по сути, представляли собой «черные ящики». В большинстве случаев эти системы использовали внутренние форматы для непосредственного управления курсами. Как правило, между системами и курсами существовало взаимно однозначное соответствие при очень ограниченных возможностях контроля пользователей. Приоритетным требованием при создании этих платформ стало предоставление и обеспечение интероперабельности конкретного контента, предназначенного для определенного курса. В это же время появился целый спектр стандартов: Dublin Core (www.dublincore.org), IMS Learning Resource Metadata (www.imsglobal.org/specifications.html), IEEE Learning Object Metadata (ltsc.ieee.org/wg12/) и ряд других, которые используются и поныне. Эти стандарты описывали контент, которым удовлетворяющие им LMS могли обмениваться на уровне ресурсов, однако такая практика не получила широкого распространения. Вместо этого стандарты обеспечивали совместимость между различными системами на уровне контента через коммуникационные каналы наподобие AICC Computer

Managed Instruction (CMI; www.aicc.org/pages/aicc_ts.htm). AICC описывал общие коммуникационные протоколы для работы с учебным контентом. В частности, были предусмотрены уведомления, посылаемые LMS и указывающие начало и конец учебного контента. Платформы первого поколения обеспечили дальнейшую эволюцию, поскольку поддерживали разделяемый контент. В качестве примера таких платформ электронного обучения можно назвать первые версии WebCT и Blackboard. [1]

Поколение 2. Платформы электронного обучения второго, или текущего, поколения (они стали появляться примерно с 1999 года) развили успех своих предшественников и начали исправлять их ошибки. К таким платформам относятся WebCT/Blackboard, Moodle и Sakai. С точки зрения эволюции электронного обучения, они обеспечили переход к реализации модульной архитектуры и показали необходимость семантического обмена. Они начали создаваться в расчете на обмен не только контентом, но и обучающими объектами, их последовательностями и информацией об учащихся. Появившиеся в это время стандарты и спецификации, такие как Shareable Content Object Reference Model (SCORM; www.adlnet.gov), IMS Content Packaging и IMS Learning Design (www.imsglobal.org/specifications.html), поддерживали способность различных платформ совместно использовать курсы или части курсов. IMS Tool Interoperability определил принципы обеспечения интероперабельности инструментов между различными системами LMS. Еще один важный аспект состоял в том, что платформы второго поколения начали поддерживать принцип «сервисов», открывая извне доступ к определенным аспектам своей функциональности. По мере повышения уровня модульности структуры становилось проще интегрировать в платформы новые функции. В некоторых случаях, например в Sakai, создаваемые Web-сервисы предоставляли доступ к ограниченному набору функций: сообщество LMS все активнее стремилось отделить контент от инструментов и уделить особое внимание информации об учащихся. Однако эти системы еще не были

полностью ориентированы на учащегося: они по-прежнему в значительной степени сосредоточены на администрировании обучения, то есть на управлении курсами. [1]

Поколение 3 (будущее). Эволюция открывает путь к платформам электронного обучения следующего поколения, что, по существу, предполагает применение сервисных оболочек к модульной архитектуре платформ. Разделение функциональности LMS и системы управления учебным контентом (Learning Content Management System, LCMS) обеспечит поддержку еще большей интероперабельности, при которой системы смогут бесконфликтно и динамически обмениваться не только контентом и сценариями обучения, но и инструментами, функциональностью, семантикой и средствами управления. Это также касается значительно более широкого спектра информации (данные пользователя, контекст, программирование, потоки заданий и управление), которую сервисы могут использовать и анализировать в платформах электронного обучения. Это даст возможность создавать настраиваемые платформы из широкого спектра сервисов для поддержки конкретных задач, динамически решаемых в определенное время. Традиционные производители LMS будут продавать не монолитные, универсальные решения, а интероперабельные платформы и разнообразные сервисы электронного обучения, предоставляя пользователям возможность выбирать нужную им комбинацию сервисов. [1]

Для перехода к системам третьего поколения необходимо разработать методику обмена информацией на семантическом уровне, что позволит не только принимать данные от аналогичных систем, но и понимать их значение. Такой уровень интероперабельности частично можно достичь используя метаданные и онтологии.

Современные системы электронного обучения включают в себя четыре компонента:

- 1 Процесс обучения.
- 1 Электронные учебные материалы.

- | Модули системы обучения.
- | Информация об обучаемом.

Для каждой из указанных компонент создаются различные стандарты, которые призваны обеспечить совместимость этих компонент, как в рамках одной системы, так и для различных LMS.

Стандарты на *процесс обучения*:

1. Educational Modelling Language (EML) – язык моделирования обучения.
2. IMS Simple Sequencing – простые последовательности.
3. IMS Learning Design - сценарии обучения.

Позволяют организовать:

- | структуру обучения;
- | роли участников процесса;
- | действия, выполняемые в ходе обучения;
- | среду, в которой проходит обучение, а также сервисы и ресурсы, доступные в среде;
- | объекты знаний, создаваемые участниками во время обучения, и доступ к ним участников;
- | передачу информации между компонентами сценария;
- | условия, выполнение/невыполнение которых влияет на прохождение сценария.

Стандарты на *электронные методические материалы*:

3. IMS Metadata – Метаданные.
4. IMS Content Packaging – Упаковка контента.
5. IMS Question & Test Interoperability – Взаимодействие тестов и вопросов.
6. AICC – Взаимодействие курса со средой обучения

Позволяют организовать:

- интероперабельность LMS на семантическом уровне
- защиту инвестиций в разработку контента

- повторное использование контента

Стандарты на *модули системы обучения*:

| IMS General Web Services – Веб службы.

| IMS Digital Repositories Interoperability – Взаимодействие цифровых репозитариев.

| OKI OSIDS (Open Service Interface Definitions) – Открытые описания интерфейсов служб.

Позволяют:

- обеспечить открытую, модульную архитектуру систем; обучения
- организовать интеграцию компонентов.

Стандарты на *информацию об обучаемом*:

• IMS Learner Information Profile – досье обучаемого.

• IMS Enterprise Services – обмен данными в организации.

• IMS e-Portfolio – электронное портфолио.

Позволяют:

| организовать сбор, хранение и передачу данных об обучаемом;

| осуществлять хранение и обработку информации о пройденных курсах;

| хранить, анализировать и представлять результаты тестирования.

Таким образом, стандартизация представленных компонентов электронного образования позволит экономить средства на разработке дополнительных интерфейсов и курсах, что в свою очередь существенно снизит стоимость таких систем и привлечет дополнительное количество пользователей и создателей контента.

Литература:

1. Сервисные платформы электронного обучения: от монолитных систем к гибким сервисам [Электронный ресурс]. – Режим доступа: <http://www.osp.ru/os /2007/07/4392612/>.

Е.В. Чернова
ГОУ ВПО «Магнитогорский государственный университет»
г. Магнитогорск

РИСКИ ДИСТАНЦИОННОГО ОБРАЗОВАНИЯ

В современном, бурно развивающемся информационном обществе специалисту требуется учиться практически всю жизнь. Если раньше можно было обучиться один раз и навсегда, и этого было достаточно для нормальной профессиональной деятельности, то сегодня необходимость «образования через всю жизнь» приводит к поиску новых методов получения знаний и технологий обучения. Дистанционное обучение открывает новые возможности для непрерывного обучения специалистов и переучивания специалистов, получения второго образования, делает обучение более доступным. Принято считать, что дистанционное образование – это процесс передачи знаний, а дистанционное обучение – это процесс получения знаний.

Цель дистанционного образования – интеллектуальное и нравственное развитие человека, продуктивно мыслящего, умеющего видеть и творчески решать возникающие проблемы. Одним из неоспоримых плюсов систем дистанционного обучения считается возможность индивидуализации процесса обучения путем составления индивидуальных планов для каждого обучаемого и автоматизации процесса контроля хода обучения. Вторым достоинством дистанционного обучения, несомненно, является его доступность и гибкость процесса: в любом месте и в любое место можно получить доступ к различным образовательным ресурсам, которые требуются для освоения того или иного курса.

Однако, на практике выяснилось, что дистанционное образование имеет некоторые специфические риски:

- 1) невозможность осуществления непосредственного контроля со стороны преподавателя за выполнением студентом учебных заданий;
- 2) невозможность контроля качества образования со стороны обучаемого;
- 3) риск мошенничества со стороны образовательного учреждения, предоставляющего возможности дистанционного образования;
- 4) проблема защиты интеллектуальной собственности;
- 5) проблема защиты информации, передаваемой в процессе обучения;
- б) длительное использование ИКТ в процессе обучения, потенциально провоцирующее возникновение различных негативных явлений в физическом, нравственном и духовном здоровье обучаемого.

Рассмотрим подробнее означенные риски.

Невозможность осуществления непосредственного контроля со стороны преподавателя в процессе выполнения обучаемым учебных заданий, и как следствие, использование чужого интеллектуального труда. Преподаватель не может отследить, кто является автором сдаваемых заданий – тот ли, кто у него обучается, или наемный работник, получивший за свою работу определенную плату. Даже при очном образовании нет гарантии, что сдаваемая курсовая работа или дипломная работа создана именно тем студентом, который ее сдает – и все благодаря целой сети компаний, предоставляющих соответствующие услуги. Точно так же очень затруднительно проконтролировать, кто решает, например, онлайн тест – тот ли, чья фамилия значится в «зачетке», или совсем другой человек.

Невозможность контроля качества образования, предоставляемого дистанционным образовательным учреждением. На сегодняшний день существует множество различных центров дистанционного образования, предлагающих программы обучения на любой вкус – от различных курсов до

полноценного диплома о высшем образовании. Однако, лишь некоторые из них дают гарантию того, что образовательные материалы соответствуют образовательным стандартам, требуемому уровню подготовки специалиста и т.п. Человеку, не знающему специфики тех знаний и умений, которые он изучает при помощи дистанционного образования достаточно тяжело увидеть, насколько ценна и весома та информация, которую он получает, для дальнейшей профессиональной деятельности по выбранной области.

Доверчивость граждан нашей страны, готовых платить деньги за услуги, при отсутствии должных гарантий ее получения, порождают риски мошенничества со стороны образовательного учреждения, предоставляющего возможности дистанционного образования: начиная от сомнительного качества образовательных программ и заканчивая «исчезновением» образовательного учреждения, вместе с деньгами за обучение.

Проблема защиты интеллектуальной собственности образовательного учреждения – любой дистанционный курс содержит пакет интерактивных материалов, разработанный данным учреждением специально для поддержки дистанционного обучения. Данный пакет предоставляется обучаемому, и нет уверенности в том, что данный обучаемый не является сотрудником, например, конкурирующих учреждений, либо иной материально заинтересованный человек. Это может привести к тому, что данный пакет материалов, претерпев минимальные изменения в названии, оформлении и т.п. появится в списках образовательных услуг других дистанционных образовательных центров, причем по более низкой цене, так как трудовые затраты будут значительно ниже, чем разработка подобного пакета самостоятельно.

Проблема защиты информации, передаваемой в процессе обучения: кража сведений об обучаемом; кража образовательных программ; заражение передаваемых файлов вредоносными программами; кража баз электронных адресов и использование их для рассылки спама и вредоносных программ;

искажение передаваемой информации (подделка результатов онлайн тестов, результатов процесса обучения – повышение баллов и т.д.).

Длительное использование ИКТ в процессе обучения, потенциально провоцирующее возникновение различных негативных явлений в физическом, нравственном и духовном здоровье обучаемого. Не возникает сомнений, что в дистанционном обучении большинство традиционных методов обучения заменяются на инновационные, с использованием интерактивных технологий и возможностей информационных технологий. И, конечно же, это приводит к тому, что обучаемый основную (а то и всю) часть времени, необходимую для освоения дисциплины курса проводит за работой с различными информационными технологиями. Поиск информации, необходимой, например, для написания реферата или курсовой, требует длительной навигации в Интернет, что очень часто перерастает в бесцельный серфинг – переход по баннерам, ссылкам, заинтересовавшим сайтам, и в итоге, мало того, что на подготовку затрачивается в разы больше времени, еще и вероятен доступ к контенту, обозначаемому как «нежелательный»: ресурсы эротического, противозаконного, провокационного, развлекательного и др. содержания. Так же, длительная работа за компьютером приводит к нагрузке на позвоночник, органы зрения, руки, органы малого таза и брюшной полости провоцируя возникновение и развитие различных болезней, начиная от близорукости и заканчивая синдромом запястного канала и нервной возбудимостью.

Таким образом, возникает острая необходимость в своевременном предупреждении возникновения рассмотренных и других вероятных рисков, для повышения надежности дистанционных образовательных учреждений и популяризации дополнительного дистанционного образования.

Ю.А.Чудинова
ООО «Корпоративные системы»
г.Магнитогорск

М.А.Шестов
ООО «Корпоративные системы»
г.Магнитогорск

ИСПОЛЬЗОВАНИЕ МУЛЬТИМЕДИЙНЫХ ОБУЧАЮЩИХ СИСТЕМ ДЛЯ ПОДГОТОВКИ ТЕХНОЛОГИЧЕСКОГО ПЕРСОНАЛА ПРОМЫШЛЕННЫХ ПРЕДПРИЯТИЙ НА ПРИМЕРЕ ОАО «ММК»

Для решения задач обеспечения безопасной эксплуатации технологических установок и оборудования предприятий, снижения потерь от незапланированных остановок по вине персонала необходимо повышать и постоянно поддерживать уровень подготовки персонала непосредственно влияющего на ход технологического процесса.

По мнению Haskett consulting inc. (НСИ): "Люди запоминают 20% того, что они видят, 40% того, что они видят и слышат и 70% того, что они видят, слышат и делают". Поэтому необходимым элементом эффективного обучения являются постоянные тренировки.

Однако там, где ошибки при обучении на реальных объектах могут привести к чрезвычайным последствиям, а их устранение - к большим финансовым затратам, наиболее эффективным является использование тренажеров.

Поэтому руководством ОАО «ММК» было принято решение о привлечение организаций с целью разработки тренажеров для стратегически важных агрегатов.

ООО «Корпоративные системы» совместно с Магнитогорским государственным техническим университетом им. Г.И. Носова по заказу ОАО «Магнитогорский металлургический комбинат» в августе 2006 года начали работу по созданию мультимедийных обучающих систем (тренажеров) для подготовки технологического персонала сталеплавильного и прокатного производств. Основной задачей, которая ставилась при

реализации проектов, было получение компьютерных программ, максимально отражающих реальные процессы сталеплавильного производства на различных агрегатах, с целью обучения и повышения квалификации технологического персонала.

В настоящее время в промышленной эксплуатации находятся следующие имитаторы-тренажеры:

- машинист дистрибутора;
- сталевар дуговой сталеплавильной печи (ДСП);
- сталевар агрегата «печь-ковш»;
- сталевар агрегата доводки стали;
- оператор главного поста управления и агрегата газовой резки слябовой машины непрерывного литья заготовок (МНЛЗ);
- оператор входной и выходной секций агрегата непрерывного горячего цинкования;
- оператор агрегата поперечной резки;
- оператор и вальцовщик сортового стана 170.

Кроме того, в разработке находятся имитаторы - тренажеры для подготовки операторов главного поста управления и агрегата газовой резки сортовой МНЛЗ, операторов постов управления стана горячей прокатки 2000, технологов стана горячей прокатки 5000.

По содержанию каждый имитатор-тренажер включает в себя четыре части обучения: устройство оборудования, пульт управления, технологический процесс, аварийные ситуации. Для каждой из этих частей есть два стандартных режима работы – «Демонстрация» и «Тестирование». Для отдельных блоков существуют дополнительные режимы работы. Система включает в себя полный набор документации, необходимый для подготовки специалистов: описание оборудования, технологические инструкции, план ликвидации аварий и другую документацию, которую должен знать обучаемый.

В режиме «Тестирование» обучаемому предлагается ряд заданий, которые необходимо выполнить. Задания, выполнявшиеся работником, а его ответы сохраняются в развернутом виде. После прохождения тестирования работник может ознакомиться с результатами и выявить слабые места в своих знаниях.

Аналогично построены другие части обучающей системы. Так, изучение пульта управления агрегата производится на основе его виртуальной модели (рис.4), которая полностью повторяет логику реальной автоматизированной системы управления технологическим процессом, включая блокировки, режим работы оборудования, основные технические характеристики, индикацию. При этом в режиме «Демонстрация» пользователь может выбрать сценарий, по которому будет осуществляться обучение. После этого на экране появляются задания, а также рекомендации по их выполнению.

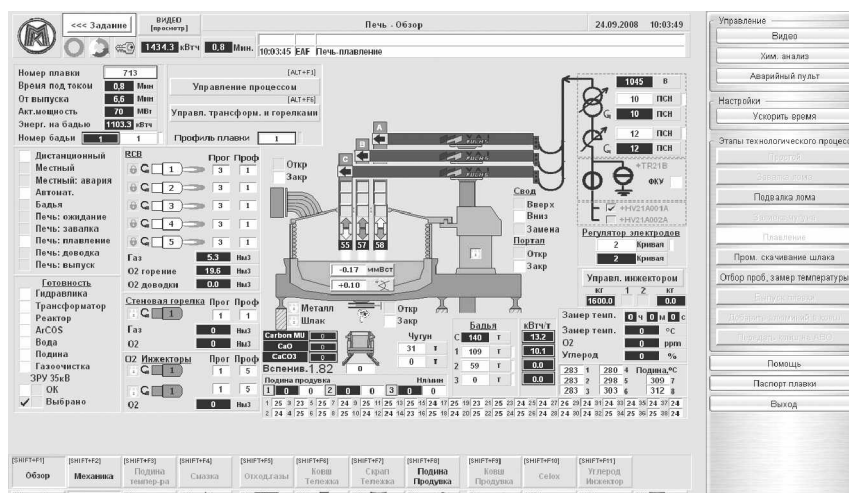


Рисунок 4 – Экранная форма «Пульт управления ДСП»

В режиме «Тестирование» работник выполняет задания, опираясь только на свои знания. Ему недоступны ни подсказки, ни иная помощь. Это позволяет объективно оценить его знания. Результаты тестирования включают в себя как выполненные задания, так и все действия пользователя. Такой подход позволяет оценить, насколько быстро и правильно было выполнено задание.

После того, как работник начнет уверенно управлять оборудованием с пульта управления, он переходит к следующему этапу – развитию и совершенствованию знаний и умений проведения технологического процесса соответствующего агрегата в условиях, максимально приближенных к реальным. Обучение технологии производится на основе разработанной Магнитогорским государственным техническим университетом математической модели технологического процесса, отображающей развитие этого процесса при заданных условиях и действиях обучаемого. При этом в режиме «Демонстрация» обучение сопровождается подробной информацией о теоретических и практических аспектах технологического процесса.

Результатом имитации технологического процесса становится паспорт плавки, а также протокол всех действий работника. Это позволяет опытному специалисту пошагово проанализировать проведение технологического процесса, выявить ошибки и дать свои рекомендации о том, как их избежать в будущем.

Последним этапом в обучении является работа в условиях аварийных ситуаций. При этом в режиме «Демонстрация» отрабатываются действия, которые следует выполнить в случае возникновения какой-либо ситуации. В режиме тестирования испытуемый сам должен идентифицировать аварию и выполнить действия, определенные планом ликвидации аварий, в допустимые нормы времени. Такой способ подачи информации позволяет выработать определенный автоматизм в действиях работника в случае возникновения аварии.

В настоящее время использование имитаторов-тренажеров осуществляется в двух цехах ОАО «ММК»: электросталеплавильном и кислородно-конвертерном. Программы установлены в учебных классах, и занятия проводятся согласно графику обучения. Кроме того, программные продукты внедряются в прокатные цеха, а также в учебный процесс

специальных учебных заведений металлургического профиля г. Магнитогорска.

Использование мультимедийных обучающих систем при подготовке технологического персонала позволило решить следующие задачи:

1. повысить качество и сократить сроки начального обучения;
2. повысить профессиональный уровень и технологическую дисциплину эксплуатационно-технологического персонала цеха;
3. обеспечить быструю переподготовку кадров внутри цеха и предприятия;
4. получить объективные оценки квалификации и профессиональных возможностей персонала цеха.

Литература:

1. Чудинова, Ю.А., Селиванов, В.Н., Колесников, Ю.А., Буданов, Б.А., Бигеев, В.А. Использование имитаторов-тренажеров агрегатов сталеплавильного производства в условиях ОАО ММК/Чудинова Ю.А. [и др.]//Металлург. – 2009. – №2. – с. 55-59.

2. Современные тренажерные технологии [Электронный ресурс]. – Режим доступа: <http://www.traintech.ru/ru/trainers/index.php>

Раздел III. Технологии Opensource

А.А. Бармин
Уфимский Государственный Авиационный Технический Университет
г.Уфа

СППР ВЫБОРА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ ВЕБ- СЕРВЕРА

Принятие управленческих решений — неотъемлемая часть работы любого менеджера. На современном этапе развития информационных технологий задача анализа текущей ситуации и принятия управленческого решения облегчается за счет использования специализированных систем поддержки принятия решений. Они помогают менеджерам решать не только слабоструктурированные и многокритериальные задачи, но принимать решения в условиях неполноты информации.

Историю развития Системы Поддержки Принятия Решений (СППР) можно проследить примерно с 1965 года. Реализовывать крупномасштабные информационные системы до 1965 года было очень дорого. Примерно в это время развитие операционной системы “IBM system-360” и других более мощных универсальных систем привело к созданию в больших компаниях систем информации для менеджмента (Management Information Systems - MIS). Они были сфокусированы на обеспечение менеджеров регулярными структурированными докладами. Большая часть этой информации поступала из систем транзакций и бухгалтерского учета.

К концу 1970-х ряд компаний и отдельных исследователей разработали интерактивные информационные системы, использующие различные модели и группы данных для помощи менеджерам в анализе проблем. Все эти разнородные системы были объединены общим понятием системы принятия решений для менеджеров. В дальнейшем оказалось, что подобные системы могут быть разработаны для поддержки всех тех, кто принимает решения на любом уровне организации. Уже в начале 1980-х гг. исследователи в области

искусственного интеллекта начали работу по созданию экспертных систем в бизнесе и менеджменте.

В начале 90-х произошел сдвиг от СППР, базирующихся на большом компьютере, к системам, основанным на компьютерных системах типа клиент-сервер. Так две технологии: поддержки принятия решений и клиент-серверная архитектура объединили свои усилия и в 1995 году появились первые СППР, доступные дистанционно через Интернет.

Безусловно, самая популярная служба современного Интернета это World Wide Web. WWW — это единое информационное пространство, состоящее из множества взаимосвязанных документов, хранящихся на веб-серверах.

Организация хостинга в Интернете и создание веб-ресурса — нетривиальная задача. Для ее решения недостаточно просто закупить оборудование и обеспечить доступ к глобальной сети. Для полноценного функционирования ресурса необходимо также наличие специализированного программного обеспечения для организации веб-хостинга.

Программное обеспечение — неотъемлемая часть любой информационной системы. Именно она позволяет реализовать полностью все аппаратные возможности компьютера.

Свободное программное обеспечение сегодня — альтернативный путь развития. Его открытость позволяет любому энтузиасту принять участие в разработке и внести свой вклад в развитие. Основным достоинством свободного программного обеспечения является его доступность, в то время как проприетарное программное обеспечение (ПО), чаще всего, сопровождается жесткими лицензионными требованиями и соглашениями.

Функциональность веб-ресурса обеспечивается тремя компонентами: операционной системой, веб-сервером и системой управления базами данных.

Операционные системы относятся к системному программному обеспечению. На сегодняшний день они представляют собой комплекс

системных управляющих и обрабатывающих программ, которые, с одной стороны, выступают как интерфейс между аппаратурой компьютера и пользователя с его задачами, с другой стороны, предназначены для наиболее эффективного использования ресурсов вычислительной системы и организации надежных вычислений

К операционным системам, устанавливаемым на веб-серверы, предъявляются повышенные требования к многозадачности, надежности и защищенности.

Нами было проведено исследование распространения операционных систем на рынке хостинговых платформ. По итогам исследования мы получили следующие результаты: на российских серверах безусловно лидирует FreeBSD (53%), на втором месте Linux (32%), Windows на третьем месте (9%), реже всего встречается Solaris (5%).

Зарубежная статистика значительно отличается. Здесь первое место занимает Linux (44%), второе — Windows (36%), третье — Solaris (12%). Российский лидер FreeBSD за рубежом практически не распространен и используется только на 7% сайтах.

В Российском сегменте Интернета наиболее распространены веб-серверы на базе свободного программного обеспечения. Распространение Windows в зарубежном сегменте можно объяснить непосредственным влиянием Microsoft на разработку и внедрение веб-серверов.

Для того, чтобы компьютер-сервер мог принимать от пользователей http-запросы необходимо специальное программное обеспечение, называемое веб-сервером.

Наибольшую популярность в Интернете имеет кроссплатформенный веб-сервер Apache, на долю которого приходится 56,5% всего Интернета. Основным достоинством Apache является гибкость конфигурации. Он позволяет подключать внешние модули для представления данных, использовать СУБД для аутентификации пользователей, модифицировать сообщения об ошибках. Сервер Apache разрабатывается и поддерживается

открытым сообществом разработчиков под сообществом Apache Software Foundation и включен во многие программные продукты, среди которых СУБД Oracle и IBM WebSphere.

Второе место занимает Internet Information Server, на долю которого приходится 30,75%. Он представляет набор серверов для нескольких служб Интернета, среди которых HTTP, HTTPS, POP3, SMTP, FTP, NNTP. IIS распространяется вместе с операционными системами Windows NT.

Таким образом, эти два продукта контролируют около 87% рынка веб-серверов. Остальные 13% приходятся на такие продукты, как HTTP File Server, lighthttpd и nginx.

Необходимым условием для создания современного веб-ресурса является наличие системы управления базами данных и возможности ее взаимодействия с веб-сервером.

В Интернете самой используемой СУБД является MySQL (используется на 29% сайтов). MySQL является решением для малых и средних приложений. Эта СУБД портирована на множество программных платформ, поддерживает механизм транзакций, кеширование запросов, возможность сортировки данных, полнотекстовый поиск, распределенность.

Второе место занимает MS SQL (24%). SQL Server используется для небольших и средних по размеру баз данных, а также для баз данных в масштабе одного предприятия. Распространяется вместе с операционной системой Windows NT Server. MS SQL поддерживает зеркализацию и кластеризацию баз данных, реализует хранимые процедуры. Эта СУБД имеет встроенные графические инструменты для администрирования.

Наименее используемой системой является DB2 (10%) фирмы IBM. Тем не менее, и про эту СУБД забывать нельзя, так в ней был впервые применен язык структурированных запросов SQL.

Задача оптимального выбора ПО при создании веб-ресурса является задачей первостепенной важности, так как выбранная конфигурация должна обеспечивать поддержку всех необходимых технологий программирования и

достаточный запас производительности, чтобы выдержать большую нагрузку со стороны пользователей веб-ресурса.

С целью выбора оптимальной конфигурации веб-сервера по заранее заданным параметрам, нами была разработана система поддержки принятия решений.

Разработанная система позволяет в интерактивном режиме подбирать конфигурацию веб-сервера по заранее заданным параметрам. К числу параметров относятся технологии программирования, возможность кластеризации и интеграции с Active Directory. Система предоставляет решения, основанные на наиболее распространенных конфигурациях веб-серверов Рунета.

СППР представляет собой динамический веб-ресурс, созданный на основе технологии PHP и реляционной базы данных. Для работы с ресурсом достаточно любого графического браузера, поддерживающего JavaScript.

Работа лица, принимающего решения заключается в выборе необходимых технологий программирования и возможностей, которые, предположительно, будут использоваться на веб-сервере. По мере выбора система автоматически предлагает решение. Стоит отметить, что решение системы не является единственно возможным при данной конфигурации. Как и при работе с любой системой поддержки принятия решений, основная работа по выбору ложится на лицо принимающее решение.

Система вырабатывает решение в три этапа. На первом выбирается веб-сервер, поддерживающий выбранные технологии программирования, или наиболее близкий по возможностям. Затем подбирается СУБД, совместимая с данным сервером и на третьем этапе выбирается ОС, на базе которой могут работать выбранный веб-сервер и СУБД. СППР учитывает возможность использования той или иной конфигурации на базе различных операционных систем и выбирает наиболее выгодный вариант. Использование проприетарного ПО выполняется только в том случае, если

невозможно получить конфигурацию с требуемыми возможностями на базе свободного.

Использование свободного ПО в качестве платформы для веб-сервера более выгодно экономически, так как лицензии, по которым оно распространяется позволяют использование ПО без необходимости отчислений производителю в случае его коммерческого использования.

Конфигурации, генерируемые системой, в целом, совпадают с общей тенденцией, используемого в Рунете программного обеспечения. Тем не менее, такие узкоспециализированные веб-ресурсы как, например, файловое хранилище или система онлайн-вещания видео предполагают большую нагрузку на программный веб-сервер. Система предусматривает генерацию решений с учетом специфики ресурса, то есть возможность распределения вычислений между несколькими компьютерами.

На основе проведенного нами статистического исследования можно сделать вывод, что программное обеспечение, используемое для организации веб-ресурсов, в среднем, одинаково как для Российского сегмента Интернета, так и для зарубежного. Более 40% за рубежом и 50% хостеров России используют свободное программное обеспечение для организации хостинга. Это в первую очередь основано на его большей надежности и более высокой скорости выхода обновлений и устранения критических уязвимостей.

Г.С. Канин
ООО «МСЦ»
г. Магнитогорск

СТАТИСТИЧЕСКИЙ АНАЛИЗ В ПРОГРАММНОМ СРЕДСТВЕ GNUMERIC

Аннотация

Сегодня статистический анализ – это инструмент, доступный не только физикам и математикам при проведении больших научных изысканий, но и рядовым экономистам, финансистам и бухгалтерам. В мире существует

огромное число программ и пакетов для статистической обработки данных. Современные табличные процессоры тоже обзавелись модулями обработки данных. Одним из представителей которых является Gnumeric.

В настоящем докладе описывается технология проведения быстрого статистического анализа (не предназначенного для серьезных научных исследований), который будет интересен экономистам, бухгалтерам, а также системным администраторам, которым в силу деятельности часто приходится обучать пользователя на рабочем месте.

Общие характеристики ряда.

Пусть имеется некий ряд данных. Задача: необходимо получить общую характеристику ряда. В Gnumeric существует целое меню, посвященное статистическому анализу: Сервис/Статистический анализ. Чтобы построить общие статистики необходимо вызвать диалоговое окно Сервис/Статистический анализ/Общие статистики (см. рис. 1). В поле «Входной интервал» необходимо ввести диапазон ячеек ряда. По нажатию на кнопку «ОК» появится новый рабочий лист с полученными результатами (см. рис. 2).

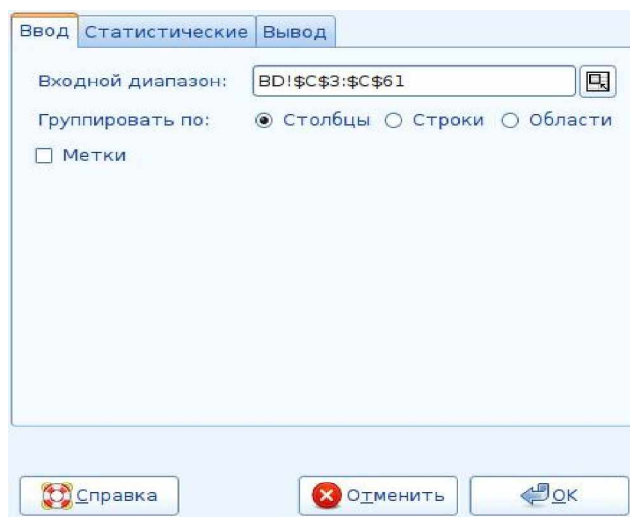


Рисунок 1 – Диалоговое окно «Общие статистики»

Gnumeric выводит стандартные характеристики ряда: количество наблюдений, среднее, СКО, дисперсия, наибольшее и наименьшее значения, мода и медиана, эксцесс.

Построение гистограммы

Для построения гистограммы необходимо запустить соответствующий диалог – Сервис/Статистический анализ/Гистограмма (см. рис. 3).

	A	B
1		Столбец 1
2	Среднее	72,3016949152542
3	Стандартная ошибка	1,0389028164851
4	Медиана	74
5	Мода	79,4
6	Стандартное отклонение	7,97996395129322
7	Выборочная дисперсия	63,6798246639392
8	Эксцесс	2,36575506267431
9	Асимметрия	-1,55340630446829
10	Диапазон	33,9
11	Минимум	48
12	Максимум	81,9
13	Сумма	4265,8
14	Количество	59
15		

Рисунок 2 – Результат вывода

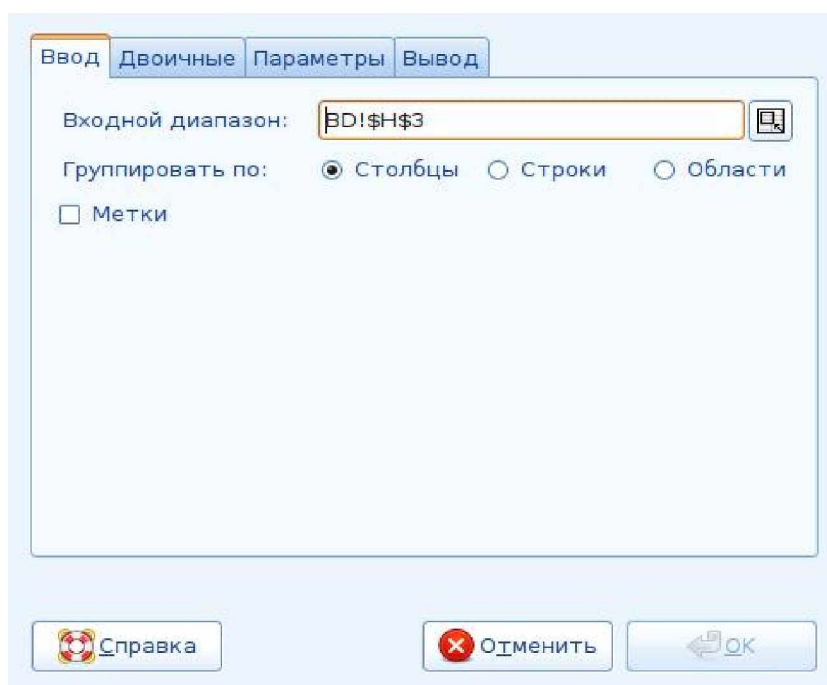


Рисунок 3 – Диалоговое окно «Гистограмма»

В поле входной диапазон необходимо указать ячейки, содержащие анализируемый ряд. Флаг «Метки» используется, когда во входном интервале содержатся названия рядов. Во вкладке «Двоичные» необходимо указать границы интервалов, для которых будет подсчитываться количество попаданий. Границы могут быть определены пользователем, тогда их диапазон необходимо ввести в соответствующее поле, или могут быть вычислены, для этого надо указать число наблюдений в ряде, наибольшее и наименьшее значения, который были определены при помощи общих

статистик. После нажатия на кнопку ОК, появляется новый рабочий лист с вычисленными значениями гистограммы (см. рис. 4).

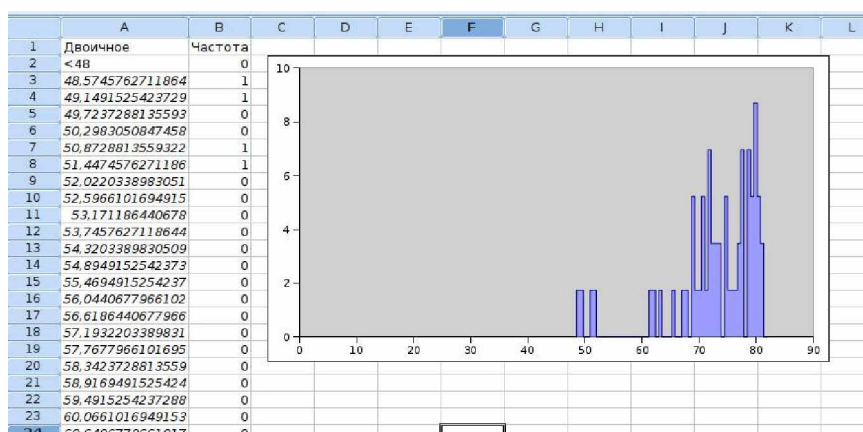


Рисунок 4 – Значения гистограммы

Для того, чтобы построить график гистограммы необходимо воспользоваться функцией вставки диаграммы, выбрав в качестве типа гистограмму. В поле «Limits» надо указать границы интервалов, в поле «Values» – полученные значения. Результат представлен на рисунке 4.

Исследование временных рядов. Выявление колебаний с помощью спектрального анализа Фурье.

Gnumeric способен строить образы Фурье, но данный алгоритм обладает некоторыми недочетами. Во-первых, алгоритм не избавляется от трендовой составляющей, во-вторых, алгоритм, как будет видно ниже, имеет некоторую погрешность. Поэтому прежде чем воспользоваться диалоговым окном Сервис/Статистический анализ/Анализ Фурье, необходимо вычесть тренд из анализируемого ряда. Самый простой способ сделать это – построить линейную регрессию.

Чтобы вызвать диалог регрессии необходимо пройти Сервис/Статистический анализ/Регрессия. Настройка происходит аналогично ранее описанным процедурам, в качестве независимой переменной надо указать временную компоненту, в качестве зависимой – значения ряда. В итоге появляется окно со значениями коэффициентов линейного регрессионного уравнения (см. рис. 5).

	степень свободы
Регрессия	1
Остатки	22
Всего	23
	Коэффициенты
Пересечение	5,84050724637681
Столбец 1	0,75652608695652

Рисунок 5 – Коэффициенты регрессии

Далее, используя полученное уравнение, можно получить значения тренда, в примере уравнение будет вида: $\hat{y} = 5,841 + 0,757t$. Далее следует из исходных значений ряда вычесть определенные оценки тренда.

В результате получим сезонную составляющую. Именно сезонную составляющую необходимо ввести в поле входного диапазона в диалоговом окне «Анализ Фурье». В результате применения алгоритма получим мнимую и действительную части образа Фурье. Чтобы вычислить значения периодограммы, необходимо найти сумму квадратов обеих частей. Для анализа результатов лучше построить график (см. рис.6). Период определяется, как:

$$p = \begin{cases} 0 & , \text{ если } t = 0 \\ \frac{N}{j} & , \text{ если } t \neq 0 \end{cases}$$

Из рисунка 6 видно, что периодограмма имеет выброс на периоде 3, но на самом деле данный ряд имеет лаг 4.

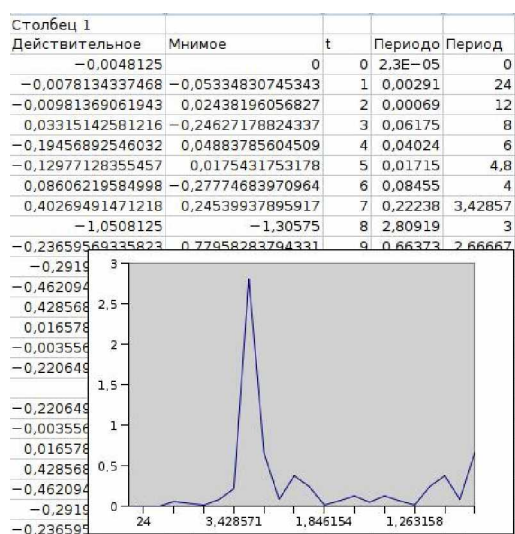


Рисунок 6 – Построение периодограммы

Подобное можно объяснить несовершенством алгоритма.

Анализ временных рядов

Для анализа временных рядов в Gnumeric существует модуль Сервис/Статистический анализ/Прогнозирование. Фактически данный модуль не позволяет прогнозировать ряды, а лишь исследовать их двумя алгоритмами: сглаживанием скользящей средней и сглаживанием экспоненциальным. Диалоговые окна приведены на рисунках 7-8. Ввод данных происходит аналогично. В результате получается новый рабочий лист со значениями экспоненциального сглаживания и сглаживания скользящей средней (в зависимости от выбранного алгоритма). Самых же прогнозных значений Gnumeric не выводит, хотя это очень просто организовать на программном уровне.

Многомерный статистический анализ

Многомерный статистический анализ подразумевает под собой построение модели, объясняющей явление. Самая простая модель – это линейная регрессионная модель. Но так как регрессия «предъявляет» ряд требований к данным, то вначале необходимо провести корреляционный анализ.

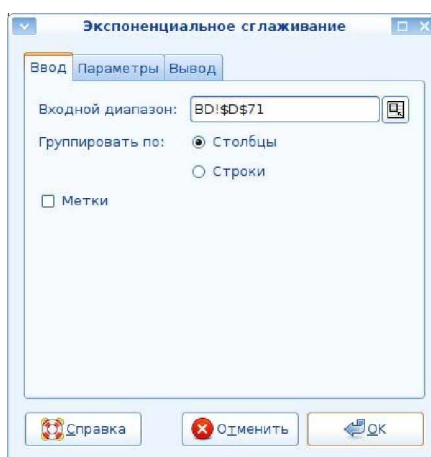


Рисунок 7 – Диалоговое окно «Экспоненциальное сглаживание»

Корреляционный анализ

Основная задача корреляционного анализа – это выявление мультиколлинеарности и снижение размерности выборки. Конечно, существуют такие методы, как гребневая или пошаговая регрессии, которые

предназначены для построения модели на мультиколлинеарных данных. В Gnumeric корреляционный анализ реализован в меню Сервис/Статистический анализ/Корреляция. Настройка алгоритма происходит аналогично анализам, рассмотренным выше. В результате появляется новый рабочий лист с корреляционной матрицей (см. рис.9).

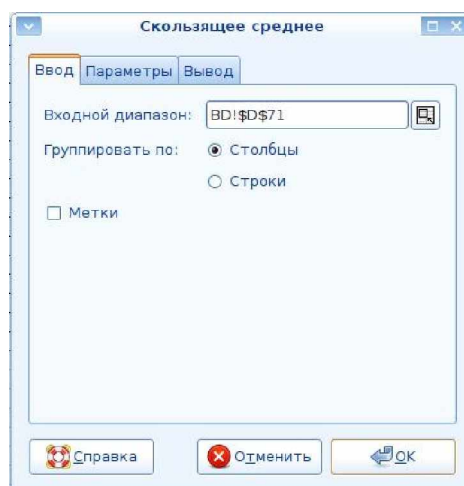


Рисунок 8 – Диалоговое окно «Скользящее среднее»

Корреляционный анализ

Основная задача корреляционного анализа – это выявление мультиколлинеарности и снижение размерности выборки. Конечно, существуют такие методы, как гребневая или пошаговая регрессии, которые предназначены для построения модели на мультиколлинеарных данных. В Gnumeric корреляционный анализ реализован в меню Сервис/Статистический анализ/Корреляция. Настройка алгоритма происходит аналогично анализам, рассмотренным выше. В результате появляется новый рабочий лист с корреляционной матрицей (см. рис.9).

	A	B	C	D	E	F	G	H	I
1	Корреляции	Y	X1	X2	X3	X4	X5	X6	X7
2	Y	1,00							
3	X1	-0,50	1,00						
4	X2	0,19	0,55	1,00					
5	X3	0,24	0,37	0,66	1,00				
6	X4	-0,11	0,74	0,65	0,53	1,00			
7	X5	0,02	0,08	0,06	-0,04	0,09	1,00		
8	X6	-0,29	0,92	0,61	0,48	0,90	0,08	1,00	
9	X7	-0,15	0,69	0,49	0,31	0,63	0,19	0,79	1
10									
11									

Рисунок 9 – Корреляционная матрица

Видно, что среди объясняющих переменных присутствует мультиколлинеарность. Не вдаваясь в подробности, исключим из выборки переменные X_4 X_6 . Теперь можно приступить к построению регрессионного уравнения.

Построение регрессионного уравнения

Описание построения регрессии приводилось в разделе «Выявление колебаний с помощью спектрального анализа Фурье». Gnumeric может построить лишь линейную регрессию. Алгоритм можно вызвать в меню Сервис/ Статистический анализ/Регрессия. В поле Y необходимо ввести объясняемую переменную, в поле X – массив объясняющих. В результате Gnumeric выводит новый рабочий лист (рис.10), где указываются коэффициенты при переменных и свободный член, который называется пересечение (пересечение с осью Y).

	Коэффициенты	Стандартная ошибка	t Stat	Значение P	Ниже 95%	Выше 95%
Пересечение	32,13	2,92	11,00	0,0000	26,27	37,99
X1	-0,44	0,05	-8,18	0,0000	-0,55	-0,33
X2	0,01	0,00	3,74	0,0005	0,01	0,02
X3	0,00	0,00	2,08	0,0424	0,00	0,01
X5	0,00	0,00	0,47	0,6374	0,00	0,00
X7	0,00	0,00	1,99	0,0512	0,00	0,00

Рисунок 10 – Значения коэффициентов регрессионного уравнения

Теперь необходимо определить значимость коэффициентов уравнения. Если значение p окажется меньше выбранного уровня значимости (0,05), то коэффициент значим, иначе – незначим. Таким образом в рассмотренном примере значимыми окажутся только коэффициенты при X_1 , X_2 , X_3 и свободный член. Тогда уравнение примет вид: $\hat{y}=32,129-0,44X_1+0,011X_2+0,0026X_3$.

Применение статистики и Gnumeric в бюджетировании

Задача: необходимо составить годовой бюджет предприятия, входящего в крупный промышленный холдинг и оказывающего услуги материнской организации. Головная компания предъявляет ряд требований. Во-первых, рентабельность должна составлять примерно 1% (небольшие колебания допускаются), во-вторых, организация в силу ряда причин может

манипулировать объемом закупок оборудования и материалов. Решение сводится к тому, чтобы сгенерировать рентабельность случайным образом, ведь стабильная рентабельность ровно в 1% может вызвать подозрения о качестве составления бюджета. На рисунке 11 представлена упрощенная структура документа. В строке 8 продублирована строка затрат с целью избежать в последующем перекрестных ссылок. Из формулы рентабельности выводится значение затрат на оборудование:

$$\text{затраты на оборудование} = (1 - \text{рентабельность}) \text{выручка} - \text{затраты. (1)}$$

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Статья	Январь	Февраль	Март	Апрель	Май	Июнь	Июль	Август	Сентябрь	Октябрь	Ноябрь	Декабрь
2	Итого затраты	19,084	20,927	23,756	25,062	24,832	25,538	28,008	24,773	26,699	27,33	25,564	26,725
3	затраты	19,084	20,927	23,756	25,062	24,832	25,538	28,008	24,773	26,699	27,33	25,564	26,725
4	затраты на оборудование												
5	Выручка	20,256	22,6624	27,589	29,6281	35,9538	35,9593	35,9977	36,0279	36,0153	35,9306	35,8810	35,8925
6													
7													
8	Затраты	19,084	20,927	23,756	25,062	24,832	25,538	28,008	24,773	26,699	27,33	25,564	26,725
9	Рентабельность												
10													
11	Текущая рентабельность	5,79%	7,66%	13,89%	15,41%	30,93%	28,98%	22,20%	31,24%	25,87%	23,94%	28,75%	25,54%

Рисунок 11 – Упрощенный вид бюджета

Чтобы сгенерировать случайным образом рентабельность пройдем в меню Правка/Заполнить/Генерация случайных чисел. Для подобных случаев лучше применять нормальное распределение. В качестве среднего возьмем планируемую рентабельность 0,01, а среднее квадратичное отклонение, которое составит 0,000(3) (или $10\% = 0,1$) определим по закону трех сигм. Во вкладке параметры укажем число переменных 1 и число выборки 12 (по числу месяцев). Полученные значения теперь необходимо проставить в строке рентабельность, а для строки затраты на оборудование прописать формулу в соответствии с (1). В итоге получим бюджет предприятия с рентабельностью около 1%.

Достоинства и недостатки статистического модуля Gnumeric.

Достоинства и недостатки статистического модуля Gnumeric приведены в таблице 1.

Таблица 1 – Достоинства и недостатки модуля Gnumeric

Достоинства	Недостатки
1. Широкий набор основных статистических алгоритмов	1. Недоработка алгоритмов
2. Наличие методов анализа временных рядов	а) Погрешности вычислений
3. Широкий набор статистический распределений в модуле генерации случайных чисел	б) Отсутствие модуля нелинейной регрессии
	в) Отсутствие средств прогнозирования
	2. Отсутствие возможности построить тренд на графике

Г.А. Лисьев
Магнитогорский государственный университет
г.Магнитогорск

В.Г. Измайлов
Магнитогорский государственный университет
г.Магнитогорск

ИНСТРУМЕНАЛЬНАЯ ПОДДЕРЖКА НАУЧНЫХ ИССЛЕДОВАНИЙ И ОТКРЫТОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

Один из авторов статьи (Г. Лисьев) руководит проектом разработки автоматизированной системы поддержки научных исследований в области экономики образования (АСПНИ) [1] в Магнитогорском государственном университете (МаГУ). Разработка инициирована научными интересами автора. Второй автор (В. Измайлов) является основным программистом проекта в составе научной студенческой группы Discovery Studio (<http://www.masu-inform.ru:8111>).

В процессе реализации проекта решаются научные, организационные, технические и дидактические задачи. Научная цель: обеспечение исследователей инструментарием для обработки информации различных типов (структурированной, слабоструктурированной, неструктурированной). Организационная цель: расширение возможностей на факультете

информатики и других факультетах (в частности, отделение социологии) МаГУ для совместной работы преподавателей и студентов над научно-техническими проектами. Техническая цель: более эффективно использовать оборудование и программное обеспечение факультета информатики МаГУ. Дидактическая цель: дополнить учебный процесс подготовки IT-специалистов опытно-экспериментальной площадкой; создать возможности для экспериментально-практической работы в обучении специалистов иных специальностей (педагогика, социология, менеджмент и др., где требуется проводить исследования с использованием массовых опросов и анкетирования).

Инициативность проекта подразумевает отсутствие (по крайней мере, на первых этапах) внешнего финансирования. Поэтому, для сведения к минимуму расходов на ПО, было выбрано открытое программное обеспечение для разработки модулей АСПНИ.

Открытая среда разработки, которая включающая в себя web-сервер Apache, СУБД MySQL и язык программирования PHP, является классическим набором инструментов для разработки web-приложений. АСПНИ представляет собой модульный программный продукт, к которому постепенно добавляются функциональные части, призванные решать определенные задачи.

К моменту написания статьи (май-июнь 2009 г.) АСПНИ размещена на web-сервере факультета информатики и доступ к «открытой» её части осуществляется через сайт Discovery Studio, раздел «Разработки DS». Также отдельные модули системы доступны в локальной сети факультета информатики. Действующие модули системы обеспечивают:

- проведение on-line и off-line опросов (анкетирования);
- доступ к созданным БД (данные ЮНЕСКО по экономике образования по странам мира);
- создание пользовательских БД;
- управление проектами (соответствие между агентами

«пользователь», «группа», «проект» и ресурсами – «файл»).

В разработке находится подсистема экспертной оценки с библиотекой алгоритмов обработки.

Модуль «Система анкетирования» вначале реализации проекта создавался как вспомогательный, для сбора и обработки статистической информации в выделенных проблемных областях. Методика сбора – опросы и анкетирование. Для создания анкет В. Измайловым был предложен и реализован подход, основанный на языке разметки [2]. Это позволило упростить создание и хранение макетов анкет, а также их дальнейшее использование (опросы, хранение результатов, экспорт данных).

В дальнейшем оказалось, что данный модуль может быть эффективно использован не только для исследовательских целей руководителя проекта, но и служить инструментом для решения образовательных задач. Например, доцент кафедры информатики П.В. Стацук использовал модуль для оценки информированности студентов факультета информатики о различных аспектах открытого программного обеспечения. А на отделении социологии исторического факультета МаГУ для студентов 3-го курса этот модуль стал инструментальной поддержкой курса «Анализ данных в социологии». Понимание того, что модуль может эффективно использоваться в учебном процессе, привело к необходимости создания локальной версии, которая может устанавливаться отдельно в сети или на сервере и работать автономно от АСПНИ. Эта задача также решена В. Измайловым. Выделение и создание в настоящее время подсистемы экспертного оценивания явилось следствием анализа эксплуатации модуля анкетирования и пожеланий пользователей (студентов-социологов, преподавателей кафедры информатики, научных сотрудников сторонних организаций – Институт динамики систем и теории управления Сибирского отделения РАН (ИДСТУ СО РАН)), а также научными интересами руководителя.

Специфика работы со студенческой научной группой обусловлена постепенностью формирования и развития необходимых компетенций у

студентов для решения поставленных задач. Поэтому в ходе совместной деятельности по реализации проекта АСПНИ приходится так согласовывать научные интересы и цели руководителя с текущими возможностями студентов, чтобы решить оптимизационную задачу: обеспечить профессиональный рост студентов и добиться продвижения в решении научных задач. Эффективность выбранной методологии подтверждается публикациями участников Discovery Studio на различных конференциях, в том числе и международной; выигранными грантами (пока только регионального уровня); возможностями студентов использовать накопленные компетенции в качестве портфолио для новых достижений (например, приглашение в Зимнюю Школу ГУ ВШЭ). Все публикации DS и некоторые статьи руководителя можно прочитать на сайте Discovery Studio.

В качестве резюме можно отметить, что реализация подобного проекта включает в себя три органические составляющие: инициатива и воля руководителя, наличие адекватного целям открытого ПО, поддержка деканата.

Литература:

1. Лисьев, Г.А. Автоматизированная система поддержки научных исследований в области экономики образования: опыт проектирования и особенности применения./Лисьев Г.А. // Открытое образование. – М.: МЭСИ, 2009. – №1. – с. 60 – 66.
2. Измайлов, В.Г. Опыт разработки в открытой среде Apache, MySQL, PHP сервисов автоматизированной системы поддержки научных исследований./Измайлов В.Г.// Материалы конференции «СТУДЕНТ И НАУКА 2009». – Магнитогорск: МаГУ, 2009.

К ВОПРОСУ ОБ ИСПОЛЬЗОВАНИИ ТЕХНОЛОГИЙ OPENSOURCE

Сложившаяся экономическая ситуация не могла не отразиться и на системе высшего образования. Одна из затратных статей любого вуза – информационная инфраструктура, которая включает в себя комплекс аппаратных и программных средств. Соответственно, возникает вопрос, каким образом можно сэкономить на покупке, к примеру, программного обеспечения (ПО)? На самом деле ответ найден, и уже достаточно давно – использование технологии OpenSource, то есть открытого программного обеспечения (ОПО). Исходный код данных программ доступен для просмотра и изменения. Это позволяет использовать уже созданный код для создания новых версий программ, для исправления ошибок и, возможно, помочь в доработке открытой программы. «Открытая» лицензия не требует, чтобы OpenSource предоставлялось бесплатно. Многие из наиболее успешных проектов ОПО, тем не менее, бесплатны.

На сайте Минкомсвязи опубликованы проекты документов, посвященных переходу органов государственной власти на свободное программное обеспечение. В документе «Методические рекомендации по разработке и приобретению программного обеспечения для использования в органах государственной власти и бюджетных учреждениях» имеется рекомендация органам государственной власти и бюджетным учреждениям отдавать предпочтение открытому ПО при выборе программных средств, за исключением случаев, когда оно не обладает необходимой функциональностью.

ОПО имеет большие перспективы в России в связи с принятием правительством и президентом РФ решений по обеспечению национальной безопасности в сфере информационных технологий на основе внедрения открытого и свободного ПО в государственные и бюджетные организации.

Весной 2007 г. на международном конгрессе-выставке «Образование без границ» АНО «Федерация Интернет Образования», Фонд содействия информатизации образования и Всероссийский интернет-педсовет представили новый проект «Свободно распространяемое программное обеспечение для образования в России». Цель проекта – заполнение информационного вакуума ОПО в образовании.

В очередной раз проблема OpenSource в жизни вообще и образовании в частности привлекла внимание широкой публики. На этот раз - благодаря усилиям Магнитогорского госуниверситета, а именно лаборатории открытых систем факультета информатики (научный руководитель – док. тех. наук, проф., советник Руководителя Росинформтехнологии А.Я. Олейников) и некоммерческому объединению пользователей свободного программного обеспечения Магнитогорской Linux User Group (руководитель И. Портнов). «Технологии OpenSource» – так назывался совсем недавно состоявшийся научно-практический семинар, призванный решить вопросы использования ОПО в вузах.

Семинар начался с доклада канд. тех. наук, доц. кафедры информатики, сотрудника лаборатории открытых систем П.В. Стащук на тему «Вопросы подготовки специалистов по OpenSource технологиям», в котором были освещены проблемы и перспективы использования ОПО в учебном процессе и информационной инфраструктуре университета в целом. Также на семинаре шла речь о таких программных продуктах как OpenOffice, Gimp, Gnumeric, Joomla и пр. С докладами выступали Г. Канин (ведущий экономист ООО «МСЦ»), П. Вязовой (web-программист КЦ «Банкир»), А. Бармин (технический директор и ведущий разработчик ООО «Сетевые технологии»), С. Шакшин (инженер ЗАО «МРК»), К. Рубан (инженер-программист факультета информатики МаГУ), К. Калугин, инженер по обслуживанию ИВТ МОУ СОШ 10, М. Сорокин (участник Linux User Group) и др.

Семинар показал, что рассматриваемое направление актуально и вызывает большой интерес, в том числе, различных организаций города с

точки зрения привлечения специалистов в области информационных технологий, владеющих навыками работы с OpenSource. Было принято решение о ежегодном проведении данного семинара, что поможет популяризации открытого программного обеспечения.

И.В. Попова
ГОУ ВПО Магнитогорский Государственный Университет
г.Магнитогорск

С.Ю. Давлеткулов
ГОУ ВПО «Магнитогорский государственный университет»
г.Магнитогорск

ЭКОНОМИЧЕСКИЕ АСПЕКТЫ OPENSOURCE

Открытый и свободный софт создавался, в значительной своей части, силами научных работников и, в том числе, для научных целей. В настоящий момент Open Source находится в сфере интересов различных компаний, в том числе, коммерческих. В свете этого представляется актуальным изучение экономических аспектов открытого программного обеспечения.

1. Обычная продажа программного обеспечения (ПО). Если ПО - OpenSource, это еще не говорит что оно бесплатно, данная программа поставляется с открытым кодом и может быть модифицирована, но продаваться как обычное коммерческое ПО.

2. Техническая поддержка. ПО может быть бесплатно и свободно распространяться (с открытым кодом естественно), но поддержка осуществляется за определенную плату.

3. Документация. Несмотря на то, что код программы открыт, разобраться в нем достаточно трудоемкая задача, особенно если речь идет о большом проекте. Компания, которая взяла за основу открытый свободно распространяемый продукт и разрабатывает компоненты для него, нуждается в документации кода продукта.

4. Благотворительность. Идеология OpenSource широко поддерживается среди пользователей, так как большая часть ПО бесплатное и довольно качественное. Поэтому довольный пользователь не откажется отблагодарить производителя.

5. Инвестиции. Если благотворительность осуществляется простыми пользователями, то инвестиции осуществляет компания-гигант (или может так же государство), мотивы данной компании очевидны: использовать части кода, которые могут модифицировать множество пользователей, в коммерческом продукте (тут уже зависит от лицензии на ПО).

6. Корпоративная оплата. Речь идет о компаниях, которым необходим качественный программный продукт, выполняющий определенные задачи. Это актуально, если аналога программы нет или есть, но он дорогой. Тогда компании вместе оплачивают создание данного продукта, а открытый код позволяет подстроить продукт под себя.

7. Платные компоненты. Компания разрабатывает только основу с малой функциональностью, а вот чтобы расширить возможности программы, необходимо приобретать модули, также компания-разработчик может писать модули и расширения под заказ.

8. Бренд. Если компания достаточно успешна и популярна, то она может зарабатывать на бренде, ставя свои фирменные знаки на различных предметах, позже ею или посредником продаваемых.

9. Продажа лицензии. Смысл в том, что сама программа открыта и свободна, но вот разрешение использования ее в качестве компонента другой требует платы.

10. Продажа носителей. Здесь имеется в виду, что пользователь может приобрести открытое ПО на носителе (CD, флешке) - даже если ПО свободно для скачивания, некоторые все же предпочитают заказывать его почтой. По существу, компания, выпускающая открытое программное обеспечение, превращается из компании-производителя в компанию, предоставляющую сервис-услуги.

И.В. Попова
ГОУ ВПО Магнитогорский Государственный Университет
г.Магнитогорск

В.В. Мурлаева
ГОУ ВПО «Магнитогорский государственный университет»
г. Магнитогорск

СРАВНИТЕЛЬНЫЙ АНАЛИЗ СРЕДСТВ МОДЕЛИРОВАНИЯ DIA И MS VISIO

Сейчас моделирование, в частности, визуальное моделирование, занимает достойное место при организации любых производственных процессов. Специфика построения схем и графиков требует разработки специальных сред, которые поддерживали бы работу с графическими примитивами и текстов, а также имели большой набор предустановленных «палитр» компонентов. В настоящее время таких программ много, но наибольшей популярностью пользуются среди лицензионных продуктов MS Visio, а среди свободно распространяемого программного обеспечения – Dia.

Целью исследования является сравнительный анализ возможностей данных средств в области визуального моделирования с учётом применения в образовательном процессе высшей школы. Для сравнительного анализа были выделены следующие критерии: назначение, варианты поставки, стоимость, общие характеристики (аппаратные и требования к программному обеспечению, скорость загрузки, локализация), поддержка шаблонов, главное окно, поддержка форматов.

Общие характеристики

Назначение

Microsoft Office Visio 2007 — программа для построения чертежей и диаграмм, помогающая специалистам сферы ИТ бизнеса визуализировать, исследовать и распространять сложную информацию, т.е. назначение Visio – графическое представление, анализ и связь комплексных данных, систем и процессов.

Dia – редактор технических диаграмм, мощный редактор векторной графики предлагает пользователям возможности по полноценному управлению процессами проектирования информационных систем. Изначально Dia планировался для построения электронных схем.

Варианты поставки

MS Visio имеет два варианта поставки: стандартный и профессиональный. Office Visio Профессиональный 2007 включает все инструменты составления бизнес-схем, доступные в выпуске Office Visio Стандартный 2007, и в дополнение к ним — ряд законченных технических решений и расширенных функций.

Dia вариантов поставки не имеет.

Как видно из таблицы 1 Dia предъявляет намного меньшие требования к аппаратным ресурсам: так, например, Dia достаточно 40 Мб оперативной памяти, что в 6 раз меньше памяти, требуемой Visio. То же самое можно сказать и о размере занимаемого дискового пространства. Относительно программного обеспечения требования обоих средств почти совпадают.

Таблица 1 – Сравнение MS Visio и Dia

	MS Visio	Dia
<i>Аппаратные требования</i>		
Процессор	Пентиум 500 МГц и выше	Пентиум 300 МГц
Объем оперативной памяти	256 Мб минимально	40 Мб минимально
Объем дискового пространства	800 Мб (1,5 Гб для установки)	42 Мб
<i>Требования к программному обеспечению</i>		
Операционная система	MS Windows 2000/XP/Vista	MS Windows 2000/XP/Vista, Linux

<i>Локализация</i>	+	+
<i>Скорость загрузки</i>	1-2 секунды	3-5 секунд
<i>Размер пустых файлов</i>	31 Кб	2 Кб
<i>Размер файлов готовых моделей</i>		
	45 Кб	21 Кб

О стоимости стоит сказать, что в связи с грядущим переходом образовательных учреждений на свободное программное обеспечение, перед ними стоит проблема поиска средств на приобретение лицензионного недешевого пакета Microsoft Office, куда входит Visio. Конечно, в такой ситуации стоит остановить выбор на бесплатном продукте Dia, распространяющимся по лицензии GNU GPL.

Локализацию, т.е. наличие русскоязычного интерфейса поддерживают оба средства. Скорость загрузки Visio в 2 раза выше и составляет 1-2 секунды. Одинаковая диаграмма с равным количеством объектов в Dia занимает меньший объем, чем в Visio в 2 раза; это связано, что dia содержит меньше метаинформации по модели.

Функциональные возможности

Моделирование

Неплохо реализованы возможности экспорта/импорта в Dia. Он поддерживает файлы *.svg, *.dxf, *.fig, а также растровые рисунки (*.bmp, *.gif, *.jpg, *.png, *.pnm, *.ras, *.tif). Свой формат программы - Dia Native Diagram (*.dia), но при необходимости созданные ею документы можно преобразовать в *.svg, *.emx, *.vdx (MS Visio), *.pdf и еще в более чем два десятка популярных форматов растровой и векторной графики. Можно сделать вывод, что Dia-просто графический редактор. Visio поддерживает некоторые графические форматы, а также форматы веб-страниц, диаграммы можно отсылать по почте прямо из программы.

Visio содержит все необходимые шаблоны и элементы для моделирования бизнес-процессов, например, такие диаграммы, как EPC, схема причинно-следственных связей, дерево ошибок, DFD, IDEF0. Dia не содержит в библиотеке примитивов для построения eEPC, диаграмму Исикавы и дерево ошибок, DFD, IDEF0 построить можно. Возможность добавления новой библиотеки или редактирование старой поддерживают и Visio и Dia.

При создании модели динамические соединения между элементами в Visio создаются автоматически, при накладывании одного объекта на другой не образуется слоев – объекты перемещаются автоматически, в Dia такого нет.

Информационное сопровождение

Справочная информация встроена как в Visio, так и в Dia, но первый может похвастаться множеством информации на сайте производителя, обучающими демонстрациями, курсами. У Dia есть учебник, но информации, представленной в нем, недостаточно для изучения особенностей построения конкретной диаграммы. И есть еще один весомый недостаток – все на английском языке.

Таким образом, MS Visio поддерживает больше типов форматов, шире библиотека элементов, а соответственно можно смоделировать бизнес-процессы в различных нотациях. Документирование и информационное сопровождение лучше опять же в Visio, а это является очень важным для разработчиков. Кроме того, в Visio поддерживаются более 30 шаблонов. Dia уступает в наличии шаблонов.

Пользовательский интерфейс

Главное окно

Структура главного окна в обоих средствах почти одинакова: окно редактирования, панель инструментов, браузер. В Dia каждое окно является независимым, т.е. перемещается отдельно от остальных окон.

В Dia для каждой диаграммы открывается отдельное окно редактирования, что очень удобно для разработчика, если существует необходимость просматривать несколько диаграмм одновременно. Вверху панели инструментов Dia расположены часто используемые примитивы. Ниже размещен раскрывающийся список библиотек элементов, при этом неудобно, что нельзя открыть несколько библиотек одновременно. Внизу панели можно выбрать цвета фона, толщину и тип соединительной линии.

Visio характеризуется тем, что каждая новая диаграмма создается на новом листе окна редактирования, для их переключения внизу расположены вкладки. При создании документа на основе шаблона в Visio, на область задач "Фигуры", играющую роль панели инструментов, подгружаются те категории графических примитивов, которые могут понадобиться в процессе создания диаграммы, плана или карты выбранного типа. Для удобства фигуры сгруппированы по тематическим категориям. Панель инструментов в Visio отсоединяется и можно поместить её в любое удобное место.

Окно просмотра (Броузер). В обоих инструментах браузер открывается только при подключении. Представлен в виде древовидной структуры объектов. Окно просмотра можно использовать для навигации по модели, отображаемой в окне редактирования. Объект выбирается щелчком мыши на объекте в браузере, он тут же отобразится в окне редактирования.

Навигация

При выборе пункта главного меню Dia, приходится все время спускаться вниз по выпадающему списку, так как выбранный пункт «закрывается» белой полосой. Оба средства обладают удобным интерфейсом, понятным любому пользователю, нужная команда находится быстро, меню не перегружено лишними пунктами.

Настройка панели инструментов

В dia настройка сводится к созданию своей библиотеки из имеющихся элементов. В Visio поддерживается создание пользовательской панели, на которой можно разместить любые инструменты.

Настройка отображения объектов

Настройка отображения в обоих средствах осуществляется в вызываемом с помощью правой кнопки мыши окне «Свойства».

Масштабирование

В обоих инструментах масштабирование представлено уменьшением и увеличением масштаба изображения, листы размечены сетчатой линией, ползунок, двигаясь по линейке, позволяет лучше ориентироваться по модели.

Особенности

Интересно, что наиболее простой способ ввода данных в диаграммы Visio - вручную. На самом же деле это не так. Для большинства пользователей Visio является лишь вспомогательной программой, которая используется наряду с Excel, Access, Microsoft SQL Server и другими решениями.

Интересно, что после размещения на диаграмме изображения аналоговых часов из библиотеки, их секундная стрелка двигается, отмеряя соответствующие мгновения, причем сами часы идут правильно.

Таким образом, выбор между dia и MS Visio зависит от множества сопутствующих факторов. Если в организации повсеместно используются продукты компании Microsoft, целесообразно приобретение MS Visio, обладающей богатыми графическими возможностями. Dia работает на любой платформе, и вполне способно справиться с задачей построения схем.

И.В. Попова
ГОУ ВПО Магнитогорский Государственный Университет
г.Магнитогорск

А.В. Яриз
ГОУ ВПО Магнитогорский Государственный Университет
г.Магнитогорск

ОБЗОР СВОБОДНО РАСПРОСТРАНЯЕМЫХ СРЕД ЛОГИЧЕСКОГО
ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ ПРОЛОГ

Пролог (Prolog) – это язык и система логического программирования, основанные на языке предикатов. В настоящий момент изучение логического программирования являются частью профессиональной подготовки студентов ИТ-специальностей. Несмотря на то, что Пролог – не единственный язык логического программирования, именно он рассматривается в большинстве практикумов.

Пролог реализован практически для всех известных операционных систем (OS для мэйнфреймов, всё семейство Unix, Windows, OS для мобильных платформ) и платформ (в том числе для Java и .NET). Кроме лицензионных продуктов также существуют «open source-среды», например Visual Prolog, YAP Prolog, GNU Prolog, SWI Prolog, Акторный Пролог и др.

Рассмотрим наиболее популярные среды с целью изучения возможности их использования в образовательном процессе вуза. Для того, чтобы описание получилось по возможности лаконичным, выделим наиболее важные критерии: компания-разработчик, платформа, на которой разрабатывается среда, надежность, законченность, основные функции, интерфейс и др.

SWI-Prolog – это мощная среда разработки с набором графических инструментов XPCЕ, распространяемая на условиях лицензии GNU GPL. Разработан компанией The SWI-Prolog Foundation.

SWI Prolog довольно популярная система, в основном благодаря удобной среде и переносимой библиотеке для создания графического интерфейса. SWI-Prolog почти как все реализации в основном следует знаменитому Edinburgh Prolog, но содержит частично реализованные особенности ISO Prolog. SWI-Prolog содержит быстрый компилятор, профилировщик, набор библиотек и удобный интерфейс для подключения Си-модулей. Он реализован для ряда UNIX-платформ таких как HP, IBM Linux, для NeXT, OS/2, Sun и Sparc.

Особенности SWI-Prolog:

- мощная система модулей;

- сбор мусора;
- поддержка юникода;
- арифметика по несвязанным целым и рациональным числам;
- многопоточность;
- гибкий интерфейс C/C++;
- интерфейс GNU Readline.

YAP (Yet Another Prolog) – высокоэффективный компилятор языка Prolog, разработанный в LIACC/Universidade do Porto и COPPE Sistemas/UFRJ. Его механизм основывается на WAM (Warren Abstract Machine) с некоторой оптимизацией для улучшения производительности. YAP является реализацией диалекта Edinburgh Prolog, но хорошо совместим со стандартом ISO Prolog.

YAP был разработан в 1985 году на языках Assembler, C и Prolog. Первая реализация обладала высокой эффективностью на машинах m68k. Более поздние версии поддерживали также архитектуры VAX, SPARC и MIPS. Работа над новыми версиями YAP преследует несколько целей:

- переносимость. Система реализована полностью на языке C. YAP компилируется на множестве платформ под управлением различных ОС
- высокая производительность. Был оптимизирован эмулятор, чтобы получить производительность, сопоставимую или большую чем в известных системах. Фактически, текущая версия YAP имеет лучшую производительность чем оригинальная, написанная на Assembler;
- надежность: YAP был проверен на множестве известных приложений для Prolog;
- расширяемость;
- законченность и совместимость с ISO Prolog;
- открытость.

YAP 4.0 и более ранние версии распространялись под лицензией, которая допускала бесплатное использование в академических средах. YAP 4.1.15 и последующие версии распространяются под лицензией Artistic.

В среде **Visual Prolog** (условно свободно распространяемая среда, лицензия нужна для Professional-версий, академические поставляются бесплатно) эффективно программируются такие приложения, как базы знаний, экспертные системы, естественно-языковые интерфейсы и интеллектуальные системы управления информацией. Высокий уровень абстракции, возможность представления сложных структур данных и моделирования логических отношений между объектами позволяют решать задачи различных предметных областей.

Визуальное создание компонентов проводится в интерактивном режиме, после чего автоматически генерируется исполняемая программа. Все необходимые файлы проекта создает эксперт приложений. Использование выбранных программистом ресурсов графического интерфейса обеспечивает эксперт ресурсов. Причем, ресурсы могут быть импортированы из динамически связанных библиотек, приложений, файлов ресурсов и других проектов Visual Prolog. Новое приложение может быть создано за достаточно короткий промежуток времени и затем последовательно расширено до конечного приложения. Такой принцип позволяет наглядно продемонстрировать переход от содержательной постановки задачи к ее логическому представлению и от логического представления к компьютерной программе.

GNU Prolog – компилятор языка Prolog, распространяющийся по лицензии GNU GPL, производительность которого сравнима с коммерческими компиляторами (по скорости совсем немного уступает SICStus Prolog). Является реализацией диалекта ISO Prolog. Некоторые особенности GNU Prolog:

- соответствует стандарту ISO Prolog (целочисленная/плавающая арифметика, потоки, динамический код, исключения);
- индексация выражений;
- множество расширений (глобальные переменные, поддержка выражений DCG, поддержка сокетов, интерфейс операционной системы);

- более 300 встроенных предикатов;
- отладчик и низкоуровневый WAM отладчик;
- сильный двунаправленный интерфейс между Prolog и C;
- компилятор, генерирующий автономные исполняемые программы;
- простой компилятор из командной строки, обрабатывающий большое количество файлов: файлы Prolog, C файлы, файлы WAM и другие.

При выборе среды реализации языка Prolog, следует учитывать множество особенностей, в зависимости от интересов пользователей. Очень важным является аспект открытости среды, так как затраты на платную реализацию достаточно большие. Также стоит учитывать удобство интерфейса и такие функции, как мощност, присутствие интерактивной среды, возможность представления сложных структур данных и моделирования логических отношений и др. Наиболее удобным и распространенной средой реализации Пролога является Visual Prolog. Важным элементом Visual Prolog являются средства разработки Web-программ, клиент-серверных приложений. Например, экспертные системы, написанные с использованием Visual Prolog, могут быть подключены к Web-страницам. Также на Visual Prolog могут быть созданы серверы баз данных или логические серверы. Визуальная среда разработки, объединяющая компилятор с текстовым редактором, комплектом инструментальных средств разработки ресурсов, экспертами ресурсов и приложений, позволяет достаточно удобно и быстро создавать приложения со стандартными пользовательскими интерфейсами.

Литература:

1. Акторный Пролог [Электронный ресурс]. – Режим доступа: <http://www.cplire.ru/Lab144/1251/09010000.html>
2. Язык программирования ПРОЛОГ [Электронный ресурс]. – Режим доступа: prologsgu.narod.ru/OI0809/NIRPRLABL.html

3. Пролог-автомат в поисковом движке: история развития языка [Электронный ресурс]. – Режим доступа: www.solarix.ru/for_developers/prolog/prolog-history.shtml · 10 КБ
4. Пролог. Генезис [Электронный ресурс]. – Режим доступа: compmuseum.narod.ru/history/prolog/prolog.html
5. Использование ТУРБО-ПРОЛОГА при проектировании баз данных и знаний [Электронный ресурс]. – Режим доступа: firm.trade.spb.ru/serp/book_prolog/Prolog3.html
6. Декларативное программирование [Электронный ресурс]. – Режим доступа: www.botik.ru/rented/lib/mupi/NewFolder/Huskell/HuskQs/dp03.html

И.В. Портнов
ГОУ ВПО «Магнитогорский государственный университет»
г. Магнитогорск

ФАЙЛОВАЯ СИСТЕМА VTRFS: РЕАЛЬНОСТЬ И ПЕРСПЕКТИВЫ

Речь пойдёт о файловой системе (ФС) нового поколения. Традиционно ФС играла значительную роль в организации Unix-систем. Во многом именно свойствами ФС определялись свойства той или иной реализации Unix. Файловая система должна хранить файлы и обеспечивать доступ к ним. При этом к ней предъявляется большое количество требований, зачастую взаимоисключающих: поддержка файлов любого размера, высокая производительность операций ввода/вывода, масштабируемость и т.д.

Давно стало ясно, что ни одна файловая система не может быть одинаково эффективна во всех случаях. Поэтому все современные реализации Unix поддерживают работу с несколькими типами ФС одновременно. Есть такое выражение: «Linux – это Unix сегодня и ядро Linux поддерживает свыше 50 (!) типов ФС».

В 2005-м году компания Sun Microsystems представила файловую систему ZFS, которая стала прорывом в области файловых систем. Из-за

лицензионной политики Sun ZFS не может быть включена в ядро Linux. Однако в 2007-м году началась разработка файловой системы нового поколения для Linux – Btrfs. Разработку оплачивает компания Oracle, однако код выпускается под лицензией GNU GPL и входит в ядро Linux начиная с релиза 2.6.29, вышедшего совсем недавно.

Приведем фрагмент интервью Chris Mason – основного разработчика Btrfs:

– Опишите Btrfs своими словами.

– Btrfs - это новая файловая система, выпускаемая под GPL, которая разрабатывается с учётом масштабируемости на очень большие объёмы носителей. Масштабируемость означает не только возможность адресовать блоки носителя, но также возможность работать с повреждениями данных и метаданных. Это означает наличие инструментов для проверки и восстановления файловой системы без отмонтирования, и интегрированную проверку контрольных сумм, чтобы определять ошибки.

– Является ли Btrfs наследницей какой-нибудь другой ФС?

– Да, всех их. Здесь много идей из ReiserFS, отложенное размещение и другие идеи из XFS. ZFS популяризовала идею, что подсчёт контрольных сумм данных может быть быстрым, и что управление логическими томами может быть лучше. Идеи по реализации управления томами пришли из AdvFS.

Итак, основные возможности, которые будут в Btrfs:

- Поддержка доступных на запись снапшотов (аналог клонов ZFS).

Кроме того, здесь можно создавать снапшоты снапшотов.

- Поддержка субтомов множественных именованных корней в одной файловой системе с общим пулом хранения.

- Поддержка сложных многодисковых конфигураций RAID уровней 0, 1, 5, 6 и 10, а также реализация различных политик избыточности на уровне объектов ФС то есть возможно назначить, к примеру, зеркалирование для какого-либо каталога или файла.

- Copy-on-write (CoW) журналирование.
- Контроль целостности блоков данных и метаданных с помощью контрольных сумм.
- Зеркалирование метаданных даже в однодисковой конфигурации.
- Полностью распределенное блокирование.
- Поддержка ACL.
- Защита от потери данных.
- Выбор хэш-алгоритма.
- Поддержка NFS.
- Флаги совместимости, необходимые для изменения дискового формата в новых версиях btrfs с сохранением совместимости со старыми.
- Резервные копии суперблока, по крайней мере по одной на устройство.
- Скоростные приоритеты для дисков.
- Гибридные пулы. btrfs старается перемещать наиболее используемые данные на самое быстрое устройство, вытесняя с него "залежавшиеся" блоки. Эта политика хорошо согласуется с появившейся недавно моделью использования SSD (Solid State Drive).
- Балансировка данных между устройствами в btrfs возможна сразу после добавления диска к пулу, отдельной командой, а не только постепенно, в процессе использования (как это реализовано в ZFS).
- Диски для горячей замены, поддержка которых появилась и в ZFS.
- Он-лайн конфигурирование RAID будет реализовано на уровне объектов файловой системы субтомов, снапшотов, файлов. Возможно будет также устанавливать некоторые параметры ввода-вывода для каталогов с наследованием этих свойств всеми дочерними объектами.
- Конвертер из ext2/3/4.

Большинство из этих возможностей уже реализованы и работают.

С точки зрения устройства ФС, можно выделить следующие основные моменты, которые делают возможными все перечисленные особенности в

сочетании с очень хорошей производительностью:

- В-деревья везде, где они имеют смысл;
- Copy-on-write везде, где это имеет смысл»
- Политика блокировок – высокая гранулярность блокировок.

В-деревья и дали название файловой системе (B-tree FS). В-деревья – это сильноветвящиеся деревья (В-деревья почему-то часто путают с двоичными деревьями, видимо, из-за буквы В, но она означает Block; у каждого узла в В-дереве обычно несколько тысяч потомков), каждый узел которых, в свою очередь, содержит большое количество записей (они обычно организуются в двоичное сбалансированное дерево; в частности, в Btrfs используются красно-чёрные деревья). Читаются и пишутся узлы В-дерева целиком, что даёт значительный выигрыш в производительности.

Copy-on-write (CoW) – это алгоритм, предназначенный для ситуаций, когда нужно создавать много похожих объектов. Рассмотрим, например, создание снапшота ФС.

Снапшот – это мгновенная копия всех данных части ФС в данный момент времени.

Реализация "в лоб" предусматривает создание копий всех файлов, что займет много времени и много дискового пространства. При использовании CoW создаётся только один новый объект – копия корневого каталога, а на всех файлах, на которые ссылается корневой, ставится специальная метка. Когда приложение пытается писать в "помеченный" каталог, ФС прозрачно для приложения делает его копию (помечая при этом все файлы в этом каталоге), и приложение пишет уже в созданную копию. Таким образом, создаются копии только изменившихся данных, и только тогда, когда данные действительно изменяются. Это позволяет сделать операцию создания снапшотов почти мгновенной даже для очень больших разделов. Это немаловажно, т.к. одно из основных требований к операции создания снапшота – атомарность, т.е. эта операция не должна прерываться (и конфликтовать) никакими другими операциями. В Btrfs CoW используется не

только при создании снимков, но и при ведении журнала и многих внутренних операциях.

Блокировки – это сущность, позволяющая избежать конфликтов при одновременном доступе к данным из разных потоков. Поток, который хочет внести изменение в некоторую структуру данных, сначала проверяет, не заблокирована ли она другим потоком; если заблокирована – ждёт, пока блокировка не будет освобождена, иначе сам захватывает блокировку, и освобождает её по окончании записи. Когда речь идёт о доступе к сложным структурам данных, возникает вопрос о политике блокировок. Нужно ли блокировать всю структуру целиком, или каждый элемент в отдельности, или элементы какими-то группами? Чем крупнее единица блокировки, тем меньше блокировок, и меньше накладных расходов. Чем мельче – тем более эффективно расходуется процессорное время, т.к. потокам приходится ждать гораздо меньше. Vtrfs стремится блокировать как можно меньшие элементы структур (но не слишком мелкие).

Ещё один пункт, связанный с блокировками, специфичен для ядра. В ядре есть два вида блокировок: spin-lock и мьютексы. При использовании spin-lock ожидающий поток "крутится" в бесконечном цикле. При использовании мьютексов – поток переходит в заблокированное состояние TASK_INTERRUPTIBLE, и "пробуждается" планировщиком автоматически при освобождении блокировки. Понятно, что мьютексы более эффективны, т.к. не тратят процессорное время на пустые циклы. Но мьютексы не могут быть использованы в контексте обработчика прерывания, т.к. в этом состоянии планировщик не работает. Значительная часть функций любой ФС может быть вызвана как из обработчика прерывания, так и в контексте задачи. Поэтому во многих функциях приходится использовать менее эффективные спин-блокировки.

Vtrfs использует новый тип блокировок, которые могут работать в обоих режимах (и их можно переключать между режимами). Таким образом, один и тот же код будет использовать мьютексы в контексте задачи и спин-

блокировки в режиме прерывания.

Ещё одна особенность Vtrfs: все структуры ФС могут находиться в произвольных местах раздела, они связаны между собой указателями. Этой особенностью обладают и некоторые другие ФС, но разработчики Vtrfs нашли ей новое применение: конвертация разделов из других ФС (сейчас реализована конвертация из ext2/3, конвертер из ext4 в разработке, теоретически можно создать конвертеры из других ФС). При конвертации структуры Vtrfs создаются в местах раздела, помеченных в исходной ФС как свободные. В Vtrfs создаётся специальный файл, в который входят блоки, занятые структурами исходной ФС. Таким образом, эти блоки оказываются помеченными как занятые. Кроме того, этот файл представляет собой образ исходной ФС, который можно примонтировать (`mount -o loop`). Это позволяет выполнить откат к предыдущей ФС.

Чтобы освободить место на диске, достаточно просто удалить файл с образом исходной ФС (возможность отката, соответственно, пропадёт).

Одна из особенностей современных ФС не так давно вызвала небольшой скандал. Дело в том, что в ядрах unix (и linux) код ФС не занимается непосредственно записью данных на диск. Данные записываются в страницы памяти, и эти страницы помечаются как "грязные и затем их сбрасывает на диск отдельный поток ядра (pdush). Так вот, при использовании современных ФС (в той новости речь шла про ext4, но теми же свойствами обладают и XFS, и Vtrfs, и многие другие) интервал между записью данных в страничный кэш и их записью на диск может достигать 150 секунд (больше двух минут). Unix традиционно пишется для хорошего оборудования. В частности, предполагается, что в любой системе, от которой нужна надёжность, применяются UPS. Поэтому большая задержка записи является не недостатком, а преимуществом: это даёт возможность разместить данные более удачно, и избежать фрагментации. А при использовании на менее надёжном оборудовании нужно просто перенастроить ядро средствами `sysctl`, чтобы заставить `pdush` срабатывать чаще.

Другая недавно вышедшая ФС для Linux - это Ext4. Она учитывает многие наработки современных ФС (экстенды, delayed allocation итп), но при этом основана на коде Ext3.

Это более продвинутая ФС, чем та же ext3, по тестам она во многих случаях даёт большую производительность и может быть рекомендована для использования на многих машинах уже сейчас. Но при этом её не назовёшь ФС нового поколения: архитектура осталась от ext3.

Btrfs сейчас в стадии experimental, разработчики предупреждают, что сейчас её имеет смысл использовать только для тестирования и экспериментов. Даже дисковый формат до сих пор окончательно не установился. Но при этом это безусловно ФС нового поколения, по архитектуре она напоминает разве что ZFS, но не старые ФС linux-ядра.

Больше всего Btrfs похожа на ZFS от компании Sun. Btrfs не поддерживает диски такого астрономического объёма, как zfs, но вряд ли это в ближайшее время будет иметь практическое значение. Зато Btrfs имеет некоторые возможности, отсутствующие в zfs: снапшоты снапшотов и скоростные приоритеты дисков, оптимизацию для ssd-накопителей.

Но ZFS уже всюду используется на production-серверах, а использование btrfs станет массовым, видимо, года через два (если предполагать, что распространение btrfs будет развиваться также, как zfs).

Измерения производительности Btrfs сейчас мало информативны, т.к. оптимизация в разгаре. Однако уже сейчас Btrfs обходит zfs по производительности некоторых операций.

Основные возможности Btrfs уже реализованы. Дисковый формат близок к стабилизации, если он и будет меняться, то не сильно. Только что завершена реализация обработки ситуации нехватки места на диске (проблема в том, что фактическая запись на диск может происходить уже после закрытия файла программой, и было не совсем очевидно, как передать ошибку записи программе). Вовсю идёт поиск и исправление других ошибок. В разработке специальный ioctl-API для поддержки транзакционного I/O

(несколько операций, объединённых в транзакцию, могут быть выполнены все или не выполнены совсем; кроме всего прочего, это позволяет минимизировать количество проверок между операциями в одной транзакции). Ближайшая задача – реализация удаления снимков, первый вариант кода уже появился в рассылке.

И.В. Портнов
ГОУ ВПО «Магнитогорский государственный университет»
г. Магнитогорск

М. Сорокин
ГОУ ВПО «Магнитогорский государственный университет»
г. Магнитогорск

РАЗРАБОТКА JAVA ПРИЛОЖЕНИЙ В NETBEANS

Технология Java разработана компаний Sun и представляет собой язык программирования Java а так же виртуальную машину Java, которая предоставляет среду для выполнения Java программ. Существует несколько реализаций Java: для разработки приложений для компьютеров, и для разработки приложений для устройств с ограниченными ресурсами (встраиваемые устройства, мобильные интернет-устройства и прочее). Каждая из них включает себя собственные наборы классов, и среду исполнения.

При разработки Java программы, исходный код компилируется в байт код, который затем выполняется виртуальной машиной Java. Благодаря этому достигается платформонезависимость кода. Написав программу один раз, ее можно запускать на любой платформе, при условии наличия виртуальной Java машины. Так же Java предоставляет возможность программирования Java-апплетов – Java-программы, встраиваемые в веб-страницу, и выполняемые в браузере. Для их исполнения так же необходима поддержка браузером.

Язык программирования Java – объектно-ориентированный. При разработке на этом языке необходимо отметить некоторые особенности. Во-

первых, сериализация объектов. Очень удобно при написании программ. Любой объект, реализующий интерфейс `Serializable` можно сериализовать стандартными средствами языка. Так, например, при написании сетевого приложения можно сериализовать произвольный объект, отправить по сети, а получатель при наличии описания этого класса, может легко получить доступ к этому объекту. Это возможно благодаря тому, что в иерархии классов Java все классы наследуются от базового класса `Object`. А для этого класса предусмотрены интерфейсы, которые и добавляют необходимую функциональность, следовательно, объявив класс, к примеру, `public class Letter implements Cloneable` можно быть уверенным, что в нужный момент мы сможем скопировать объект данного класса без написания дополнительного кода.

Программисты C++, знают насколько важно вовремя освободить память. Всем известна и проблема утечек памяти в программах. В Java же все по-другому. Программист не задумывается об освобождении памяти, в Java для этих целей используется сборщик мусора, виртуальной Java машины. Кратко это можно описать так. Когда запускается Java программа, `JVM` в отдельном потоке запускает сборщик мусора, который, периодически получая управление, отслеживает использование объекта, и, если объект не используется, он удаляется и память освобождается. Разумеется, в таком случае эффективность работы с памятью полностью зависит от реализации сборщика мусора, но в программах, оперирующих объектами большого объема, программист может использовать принудительный вызов сборщика мусора, а так же функцией `dispose()` самостоятельно удалить объект.

Так же хотелось бы отметить механизм исключений. В Java – все является объектом – и исключительная ситуация, не является исключением. В иерархии классов присутствует базовый класс `Exception`. Если в программе возникает исключительная ситуация, создается объект данного класса (или класса, наследуемого от него), который прекращает выполнение текущего блока кода. Исключение немедленно передается обработчику исключений.

Более того, любое исключение в Java должно иметь обработку, иначе код не будет скомпилирован. Если в определенном методе класса, возникает исключительная ситуация, она должна быть либо обработана в данном методе, либо если обработка не реализуется в методе, то в описание метода добавляется признак того, что метод может породить исключительную ситуацию. В таком случае, вызов функции, без обработчика исключений будет невозможен. Что бы различать типы исключительных ситуаций существуют различные подклассы класса Exception. В обработчике же класса можно получить достаточную информацию о случившейся проблеме. Так, Exception e; e.printStackTrace() – напечатает последовательность функций вызвавших исключительную ситуацию или e.getMessage() – напечатает локализованное сообщение о случившейся проблеме.

NetBeans – среда разработки построенная по принципу расширяемости от компании Sun. Существуют версии для разработки на языках Java, php. Так же NetBeans поддерживает plugin'ы (см. рис.1).

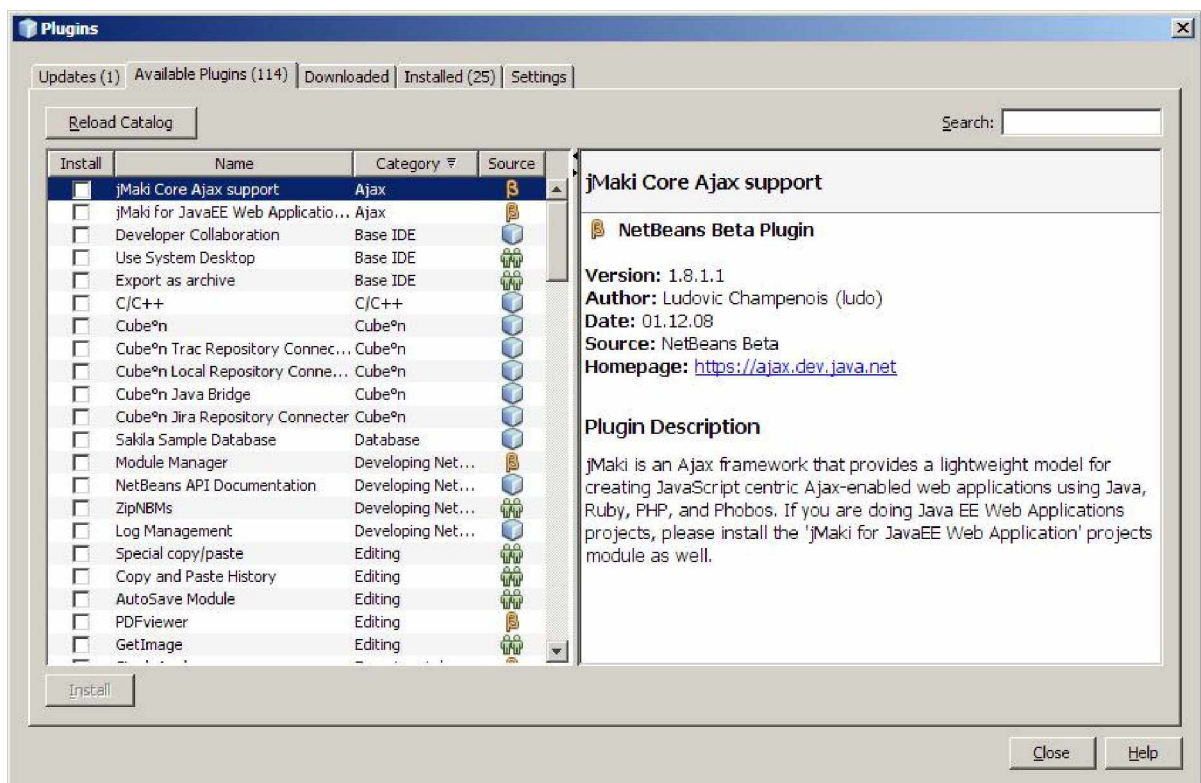


Рисунок 1 – Диалог управлением plug-in-ами

Так, например, можно установить поддержку языка с/с++, просто установив соответствующий plug-in. Итак, рассмотрим стандартную поставку NetBeans для разработки на языке Java.

Разработчик получает в свое распоряжение: удобный редактор кода, с автодополнением, и автозавершением кода, редактор форм, (используются классы `Javax.swing*`), отладчик, профайлер. Так же имеются дополнения для подключения к системам контроля версий.

Для перехода в режим дизайнера форм, необходимо наличие созданной формы в проекте. В случае ее отсутствия, ее можно создать, затем переключение между режимами дизайна и редактирования кода осуществляется соответствующими кнопками. В режиме редактора форм, становится доступной палитра компонентов, доступных для размещения. Так же программист может добавить в палитру собственные компоненты. Для этого, нужно выбрать необходимый класс, затем выбрать в меню пункт «Tools», и затем выбрать пункт «Add to palette». Редактор форм, позволяет задавать параметры создаваемого объекта визуального компонента: можно задать атрибуты доступа, а так же отредактировать код инициализации.

NetBeans располагает удобными средствами для отладки приложений. Присутствуют break-point'ы, значение в конкретный момент можно узнать простым наведением мыши на нее в редакторе кода.

Профайлер – еще одно средство в арсенале программиста. Если отладчик позволяет контролировать выполнение программы, то профайлер позволяет видеть время выполнения, и затраты ресурсов программы. Причем визуально можно отследить, сколько времени приходится на объекты каждого класса. Так же отслеживается процентное время выполнения, относительно общего времени исполнения. Так же отслеживается использование памяти. Таким образом, можно выявить на этапе разработки слабые места в производительности приложения, и спроектировать код для рационального использования ресурсов системы.

ПРИМЕНЕНИЕ СИСТЕМ ОБНАРУЖЕНИЯ ВТОРЖЕНИЙ НА ПРИМЕРЕ ПРОГРАММЫ TRIPWIRE

Сегодня, когда сайтов с описаниями взломов, а также статей, эксплоитов и программ на эту же тему, появилось очень много, намного участились попытки взлома серверов. Любой пользователь, вообразивший себя хакером, может испробовать на Вашей системе написанный кем-то эксплоит и получить контроль над Вашим сервером. Для того чтобы узнать про атаки, или просто быть в курсе всех событий, происходящих на сервере, многие администраторы ежедневно просматривают журналы регистрации, но когда журнал, например веб-сервера Apache, за день вырастает на 10 мегабайт, то приходится автоматизировать процесс контроля безопасности сервера. Одним из решений являются системы обнаружения вторжения (Intrusion Detection System, IDS), эффективность которых за последние несколько лет значительно возросла, и теперь они являются неотъемлемой частью любой сетевой защиты.

Система обнаружения вторжений – программное или аппаратное средство, предназначенное для выявления фактов неавторизованного доступа в компьютерную систему или сеть либо несанкционированного управления ими.

Теоретически, о важности и необходимости систем обнаружения вторжений начали говорить со времен, когда ARPAnet вышел за пределы военных и исследовательских институтов США.

Первые практические наработки в области IDS появились чуть более двух десятилетий назад, и заметно уступали по функциональности современным аналогам. основополагающими документами при их построении считаются труды Джеймса Андерсона «Мониторинг угроз

компьютерной безопасности» и Дороти Деннинг «О модели обнаружения вторжения» (опубликованы в 1980-х гг.).

На сегодняшний день IDS принято классифицировать по нескольким параметрам, к числу которых относятся способ сбора информации, метод анализа информации и способ реагирования на угрозы.

Все системы обнаружения вторжений можно разделить на 2 большие категории:

- Network Intrusion Detection System (NIDS) – системы основанные на анализе сетевого трафика, передаваемого и получаемого защищаемой системой.
- Host Intrusion Detection System (HIDS) – основаны на анализе локальной информации (CRC файлов, журналов и т.д.).

В общем виде принцип действия NIDS можно описать так: весь трафик анализируется на наличие некорректных пакетов, либо пакетов содержащих вредоносные данные, и если такие пакеты находятся, то выполняются различные сигнализирующие действия (выводится сообщение на консоль, посылается почта, пишется сообщение в журнал, и.т.д.) и происходит блокировка пакета (активные IDS).

Среди сетевых IDS выделяют:

- IDS основанные на анализе протокола (выявляющие пакеты несоответствующие стандартам), и
- IDS основанные на анализе сигнатур (в пакетах ищется сигнатура атаки – строка (образец) указывающая на принадлежность данного пакета к вредному трафику.

Примеры NIDS: Snort, Ipllog, Libnids, и т.д.

Целью хостовой IDS является слежение за всеми событиями, происходящими в системе и проверка их на соответствие модели безопасности. В то время как сетевая IDS отслеживает проходящие сетевые пакеты, хостовая IDS проверяет, какая программа к каким ресурсам обращается и может обнаружить, что, например, текстовый процессор начал

менять системную базу паролей. Хостовая IDS просто ведет наблюдение за текущим состоянием системы, за хранимой информацией (как в оперативной памяти, так и в файловой системе), за данными системных логов и проверяет, насколько это состояние соответствует «нормальному». Можно сказать, что хостовая IDS – это агент, наблюдающий за тем, чтобы никто не смог нарушить (снаружи или изнутри системы) политику безопасности, установленную операционной системой.

Примеры HIDS: Tripwire, OSSEC, rkhunter, Aide, и т.д.

Очевидно, что NIDS способны значительно раньше обнаружить попытку вторжения и имеют возможность его предотвратить, но это касается действий исключительно сетевых пользователей. Кроме того, NIDS требуют тонкой настройки, достаточно требовательны к ресурсам, «болеют» ложными срабатываниями и подвержены некоторым видам атак (переполнение потокового буфера, фрагментация пакетов, корректировка TTL и т.д.).

В свою очередь HIDS позволяют установить, были ли осуществлены противоправные действия в системе и в зависимости от результата информировать администратора системы, но при этом – факт взлома уже состоится!

HIDS TripWire разработана доктором Южином Спаффордом (Eugene Spafford) и Жене Кимом (Gene Kim) в университете Пурде (Purdue University) в 1992. В 1997 все права переданы коммерческой организации Tripwire, Inc. В 2000 году появилась специализированная версия под Linux, которая была лицензирована под GPL.

Суть работы Tripwire заключается в создании базы данных эталонных атрибутов важных файлов, которые предполагается отслеживать. Таким образом, в любое время можно сравнить текущие атрибуты с эталонными, чтобы узнать, изменилось ли что-нибудь.

Установка TripWire

Tripwire включен во многие популярные дистрибутивы, хотя присутствует возможность установки программы из исходных кодов, которые доступны для скачивания на официальном сайте проекта (<http://www.tripwiresecurity.com/>).

В процессе установки необходимо задать парольную фразу, на базе которой создаются файлы ключей (`/etc/tripwire/site.key` и `/etc/tripwire/host-local.key`).

Создание и редактирование файла настроек

Конфигурационный файл: `/etc/tripwire/tw.cfg`, хранится в закодированной и подписанной форме. Преобразовать его в текстовый формат можно командой `twadmin --print-cfgfile > /etc/tripwire/twcfg.txt`, после чего отредактировать любым текстовым редактором, закодировать и подписать командой `twadmin --create-cfgfile [--no-encryption] /etc/tripwire/twcfg.txt`, а текстовый файл немедленно удалить.

Конфигурационный файл содержит настройки путей, методов нотификации, глубины отчета и т.д.

Создание и редактирование файла политик

Политики: `/etc/tripwire/tw.pol`, хранятся в закодированной и подписанной форме. Преобразовать файл политик в текстовый формат можно командой `twadmin --print-polfile > /etc/tripwire/twpol.txt`, после чего отредактировать любым текстовым редактором, закодировать и подписать командой `twadmin --create-polfile [--no-encryption] /etc/tripwire/twpol.txt`, а текстовый файл немедленно удалить. После этого требуется создать БД заново.

Файл политик содержит правила (rule), которые задают для системных объектов (файлов или директорий, файлы внутри директории наследуют указанное правило, однако, рекурсия не пересекает границу файловой системы) наборы отслеживаемых свойств (не более одного правила для объекта):

- полное-имя-объекта -> маска-свойств [(атрибуты)];

- !полное-имя-игнорируемого-объекта;

Ниже приведены несколько свойств обрабатываемых Tripwire (для ознакомления с полным списком свойств, рекомендуется обратиться к официальной документации):

- p – права доступа;
- n – число жестких ссылок;
- u – uid;
- g – gid;
- t – тип файла;
- s – размер;
- и т.д.

На базе свойств можно создавать переменные, которые впоследствии используются для определения объектов. Ниже перечислены переменные по умолчанию:

- `ReadOnly = +pinugsmtdbCM-raclSH` (предполагается, что содержимое файлов не изменяется).
- `Dynamic = +pinugtd-rsacmb1CMHS` (предполагается частое изменение содержимого файлов, например /home).
- `Growing = +pinugtdl-rsacmbCMSH` (например, журналы).
- `IgnoreAll = -pinusgamctdrblCMSH` (проверять лишь наличие или отсутствие файла).
- `IgnoreNone = +pinusgamctdrbCMSH-l` (используется для дальнейшего конструирования собственной маски: `$(IgnoreNone)-ar`).
- `Device = +pugsdr-intlbamcCMSH` (для файлов, которые нельзя открывать).

Например, для критических данных можно определить следующую переменную: `SEC_CRIT = $(IgnoreNone)-SHa; # Критические файлы`

Теперь на базе данной переменной можно создать правило:

```
(emailto = admin@openna.com, rulename = "Critical system boot files",  
severity = 100)
```

```
{  
/boot -> $(SEC_CRIT);  
!/boot/System.map;  
!/boot/module-info;  
}
```

Как видим, при создании правила, дополнительно устанавливаются параметры нотификации, в случае проблем.

Инициализация базы

Запускать Tripwire в режиме создания образа желательно на «чистой» системе, установленной из официальных источников.

Синтаксис для перехода в режим инициализации базы данных:

```
[root@redhawk /]# tripwire -m i -v
```

Ключ -m определяет режим выполнения, в данном случае i означает инициализацию. Ключ -v задает расширенный вывод, чтобы можно было посмотреть, что происходит. Tripwire определяет все файлы, заданные в файле политики, создает базу данных в каталоге ./database и шифрует ее с помощью site.key.

Проверка целостности файлов

Это основной режим выполнения программы Tripwire после ввода в эксплуатацию. В этом режиме текущие атрибуты определенных файлов сравниваются с атрибутами в базе данных Tripwire. Формат запуска в этом режиме следующий:

```
[root@redhawk /]# tripwire -m с маршрутное_имя
```

Обновление базы и политик

```
[root@redhawk /]# tripwire -m u
```

```
[root@redhawk /]# tripwire -m p текстовый_файл_политики
```

Как видим, настройка HIDS Tripwire проста и не займет много времени, при этом эффективность использования значительно выше ручного анализа данных.

Отметим, что для организации качественной защиты от вторжений, важно комбинировать несколько подходов и использовать как хостовые, так и сетевые IDS.

С. Шакшин
ЗАО «Механоремонтный комплекс»
ОАО «Магнитогорский металлургический комбинат»
г.Магнитогорск

О ПРОЕКТЕ CENTRIX

Современный системный администратор обычно выполняет широкий спектр задач. Можно сказать, что самой распространенной обязанностью администратора является настройка пользовательских рабочих мест, а именно установка и настройка программного обеспечения. При этом, как правило, на большей части управляемых машин устанавливаются однообразные наборы программ. Для автоматизации процесса установки и настройки создано немало инструментов. Самым ярким примером является домен Windows Server. Но, к сожалению, ни платные, ни бесплатные инструменты не дают главного – универсальности.

В сети любого крупного предприятия можно встретить компьютеры с разными операционными системами, которые предъявляют свои уникальные требования к системе управления. Кроме того, не все программные продукты предлагают интерфейсы для удаленной настройки. Именно поэтому спектр программного обеспечения (ПО), которым можно управлять удаленно, существенно ограничен.

Таким образом, начальной целью проекта Centrix является создание максимально универсального и гибкого инструмента для удаленного управления пользовательским ПО. По ходу развития проекта, к основной цели добавились новые задачи и функции:

1. создание, хранение и применение пользовательских профилей (наборов конфигураций различного ПО);
2. выполнение команд на управляемых машинах с возможностью выполнять одновременно одну команду на группе машин;
3. мониторинг различных событий и состояний на управляемых машинах;
4. максимальное избавление от ручного конфигурирования пользовательских машин;
5. достижение максимальной отказоустойчивости системы – при выходе из строя любого из серверов система должна продолжить работу с минимальными отклонениями от нормального функционирования;
6. достижение открытости системы и, как следствие, масштабируемости за счет внешних расширений;
7. создание кроссплатформенного решения.

Базовая архитектура

Система Centrix имеет клиент-серверную архитектуру. Каждый сервер может работать в двух основных режимах: Master и Relay. Сервер в режиме master является «главным» сервером сети Centrix и на нем находится главное хранилище профилей. Серверы в режиме Relay являются своего рода шлюзами и служат для снижения нагрузки на межсетевые каналы связи и для обеспечения отказоустойчивости. Сервера организуются в древовидную структуру, «корнем» которой является главный сервер. Пользовательские машины в данной структуре будут являться «листьями» и подключаться либо к центральному серверу, либо к релейам. Для управления всей этой структурой у администратора должен быть инструмент – консоль. С помощью консоли администратор подключается к любому из серверов системы и производит необходимые операции.

Всё сетевое общение в системе Centrix планируется реализовать на базе XML-сообщений внутри TCP-соединений.

Слой взаимодействия

На логическом уровне все взаимодействие сосредоточено между центральным сервером и компьютером-клиентом, а сервера-релеи выступают в роли шлюзов и кеш-хранилищ. Это взаимодействие можно разделить на три слоя:

- Слой конфигураций – все общение касательно пользовательских профилей: запросы, ответы, первоначальная привязка.
- Слой команд – удаленное выполнение команд на пользовательских компьютерах.
- Слой мониторов – сообщения-события, сгенерированные на пользовательских компьютерах специальными модулями-мониторами.

Модули на стороне клиента

Эти слои реализуются посредством модулей-расширений на стороне клиента. Сам клиентский модуль не занимается настройкой чего-либо. Он только общается с сервером, получает команды и профили и отдает их модулям, которые выполняют соответствующие действия. Таким образом, можно создавать расширения под свои нужды. Каждый модуль фактически является самостоятельной программой, которая запускается клиентским модулем и которой могут передаваться параметры. Модули делятся на три группы (для реализации разных слоев) и отличаются методом запуска и поведением. Так модули-конфигураторы запускаются при получении конфигурации для соответствующего им ПО, выполняют свои действия по настройке и завершают свою работу. Аналогично модули команд, получают команду, выполняют ее и завершаются. Модули-мониторы запускаются при старте клиентского модуля и завершают свою работу вместе с ним или при остановке системы. Эти модули через специальный интерфейс генерируют сообщения о наступлении некоторых, отслеживаемых ими, событий.

Серверные компоненты

Серверная часть так же состоит из нескольких компонент, но их структура отлична от структуры клиентской части.

1. Главным серверным компонентом является ядро – сетевой модуль, который взаимодействует с другими серверами и клиентами. Ядро обрабатывает принятые сообщения и выполняет действия по обслуживанию клиентов на основании этих сообщений.

2. Центральное хранилище конфигураций служит для хранения профилей всех пользователей сети Centrix. Хранилище может быть реализовано в виде БД, в виде файловой структуры и т.п. Та или иная реализация подключается путем выбора нужного back-end'а для хранилища.

3. Хранилище модулей для клиентов (репозитарий) может быть выполнено в виде файлового сервера FTP, Samba, в виде HTTP-сервера или в любой другом виде, доступном как для клиентских машин, так и для серверов.

4. Процессор событий – модуль, обрабатывающий поступающие от клиентских мониторинговых модулей события. Администратором составляется таблица соответствия событие-обработчик, сходное по функционалу с таблицами iptables. Обработчики являются внешними приложениями или скриптами и вызываются при срабатывании записи в таблице соответствия.

Все эти серверные компоненты могут быть расположены на одном физическом сервере, а могут быть разнесены на несколько машин.

Отказоустойчивость

В целях реализации отказоустойчивости системы предлагается добавление еще одного режима функционирования сервера – Backup Master. В этом режиме сервер ведет себя как обычный релей, но при запуске объявляет себя резервным Master'ом. Таким образом, центральный сервер будет реплицировать все изменения конфигураций на резервный. В случае, если центральный сервер аварийно перестал функционировать, все остальные сервера будут обращаться к резервному, как к основному, а все изменения, вносимые администратором будут сохраняться на резервном сервере, а после восстановления работы центрального сервера будут реплицированы на него.

В случае, когда проблема не с центральным сервером, а например, пропало сетевое соединение между одним из сегментов сети и главным сервером, поведение системы выглядит иначе. В этом случае изолированный сервер будет вести себя как обычный релей. Если в это время администратор вносит изменения вне этого изолированного сегмента, эти изменения будут храниться на главном сервере и станут доступны в этом сегменте лишь после восстановления связи. Если же администратор оказался в изолированном сегменте и вносит изменения, то они сохраняются на самом «верхнем» доступном релее и реплицируются на главный сервер после восстановления связи.

В этих механизмах необходимо предусмотреть алгоритм разрешения конфликтов между конфигурациями. Этот механизм основывается на времени изменения конфигурации. В результате слияния конфигураций, действующей будет та, которая позже изменена.

Текущее состояние разработки

На данный момент проект Centrix находится на ранней стадии, в рамках которой осуществляется проработка концепции и сбор идей. Проект открытый и все желающие могут присоединиться к нему в роли разработчика или просто поделиться идеями и пожеланиями.