



МАГНИТОГОРСКИЙ
ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ ИМ. Г.И. НОСОВА»

ТЕОРИЯ И ПРАКТИКА ПРИМЕНЕНИЯ СВОБОДНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ



Сборник материалов Российской молодежной
конференции с элементами научной школы,
18-22 мая 2016 г



МАГНИТОГОРСК, 2016

*При поддержке
Российского Фонда фундаментальных исследований*

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Магнитогорский государственный технический университет
им. Г.И. Носова»

ТЕОРИЯ И ПРАКТИКА ПРИМЕНЕНИЯ СВОБОДНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

*Сборник материалов Российской молодежной конференции
с элементами научной школы
18-22 мая 2016 г*

Под общей редакцией
Гавриловой И.В., Назаровой О.Б.

Магнитогорск
2016

УДК 004.4
ББК 32.81+32.97

Общая редакция: канд. пед. наук, доц. И.В. Гаврилова
канд. пед. наук, доц. О.Б. Назарова

Технический редактор: канд. пед. наук, доц. Л.В. Курзаева

Теория и практика применения свободного программного обеспечения: сборник трудов участников Российской молодежной конференции с элементами научной школы, 18–22 мая 2016 г. / под общ. ред. Гавриловой И.В., Назаровой О.Б. – Магнитогорск: Изд-во Магнитогорск. гос. техн. ун-та им. Г.И. Носова, 2016. 223 с.

ISBN 978-5-9967-0842-0

В сборнике представлены доклады участников Российской молодёжной конференции с элементами научной школы «Теория и практика применения свободного программного обеспечения», отражающие результаты научно-исследовательской работы участников.

Материалы сборника адресованы преподавателям вузов, учителям школ, работодателям, ИТ-специалистам, вендорам, студентам и аспирантам.

Публикация выполнена при финансовой поддержке Российского фонда фундаментальных исследований в рамках проекта № 16-37-10124-моб_г.

Дизайн обложки: Э.Н. Коршунов

УДК 004.4
ББК 32.81+32.97

ISBN 978-5-9967-0842-0

© Магнитогорский государственный
технический университет
им. Г.И. Носова, 2016

СОДЕРЖАНИЕ

<i>Петеляк В.Е., Гаврилова И.В., Давлеткиреева Л.З.</i> Российская молодёжная конференция с элементами научной школы «Теория и практика применения свободного программного обеспечения»	6
РАЗДЕЛ 1. СВОБОДНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ В ОБРАЗОВАНИИ И НАУКЕ	14
<i>Дегтярева К. С., Мухеева А. Р.</i> Сюжетно-ролевое обучение работе в среде программирования Scratch	14
<i>Гусева Т.Ф.</i> Опыт использования свободного программного обеспечения при подготовке ИТ - специалистов	19
<i>Емельянова Ю. А., Сироткина А. С.</i> Использование Scratch для разработки интерактивных дидактических средств	24
<i>Казакова Т. В.</i> Реализация системы автоматической оптимизации в Scilab	30
<i>Карманова Е.В.</i> Свободное программное обеспечение в электронном обучении	34
<i>Козлова Е.С.</i> Построение онтологической модели организации образовательного процесса в системе Protege	41
<i>Курзаева Л. В., Соколова А. А., Кириллов Д. В.</i> Онтологическое моделирование содержания профессиональных компетенций в среде «Protege»	49
<i>Предеина О. С., Аркатова К. А.</i> Планирование внеурочного занятия по теме «Информационная безопасность» с использованием среды Scratch	58
<i>Окжос К.М., Шабалина М.И.</i> Сравнительный анализ информационных систем для автоматизации деятельности редакции научного журнала	63
<i>Пуляевская А. М., Вислова К. В.</i> Интерактивное видео: инструменты и возможности применения	71
<i>Рабовская М. Я., Махнева В.В.,</i> Индивидуальная траектория студента как часть автоматизированной системы составления расписания в вузе	78
<i>Фукалов Д.С.</i> Разработка электронного журнала научных статей кафедры	84

Вычислительной техники и инженерной кибернетики Уфимского государственного нефтяного технического университета для публикации в сети интернет на базе Open Journal Systems	
РАЗДЕЛ 2. СВОБОДНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ В БИЗНЕСЕ	91
<i>Алейникова М.А., Подколызина Л.В., Махмутов Р. Р.</i>	
Применение систем с открытым исходным кодом для проектирования технологического процесса механической обработки детали	91
<i>Вахрушев В. И., Гусева Т. Ф., Дьяков В. А.</i>	
Разработка моделей стоимости недвижимости в Нейросимулятор 2.0	96
<i>Кириллов Д.В., Соколова А. А.</i>	
Автоматизация процесса управления бизнес-процессами предприятия	102
<i>Мавлянов Т. Б.</i>	
Диалоговая система "MPRIORITY 1.0"	108
<i>Севостьянова А. В., Столяров А И.</i>	
Обзор пакета Xelopes и его основных возможностей	112
<i>Ягудина Р. Р.</i>	
Внедрение системы управления бизнес-процессами для реинжиниринга сельскохозяйственного предприятия	115
<i>Ягудина Р. Р.</i>	
Построение онтологической модели базы знаний интеллектуальной информационной системы подбора кадров	119
РАЗДЕЛ 3. СВОБОДНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ В ПРОГРАММНОЙ ИНЖЕНЕРИИ	127
<i>Беков Б.Г., Шибанин А.С., Сергеев К.А.</i>	
Web-интерфейс вывода статистики системы учета студентов	127
<i>Галимова Е. Ю.</i>	
Анализ алгоритма принятия решения об автоматизации тестирования программного продукта с применением свободного программного обеспечения Selenium	133
<i>Журавлев А. Е.</i>	
Особенности внедрения внутренних информационных разработок в действующую проприетарную информационную систему университета	138
<i>Конькова Д. С.</i>	
Сравнительный анализ открытых систем разработки онтологий DOE и PROTÉGÉ	146

<i>Сапегина В. С.</i> Построение онтологии «Автодилеры» в программной среде Protégé 4.2	151
<i>Столяров А. И., Севостьянова А. В.</i> Обзор открытых оболочек для создания экспертных систем	155
<i>Столяров А. И., Севостьянова А. В.</i> Создание онтологии «Лекарственные средства» в системе Protege	161
<i>Капцан А.В, Черномырдин А. В., Матвеев В., Чернов В.О.</i> Технологическая платформа RadixWare	166
<i>Сорокин М.О.</i> Автоматизация тестирования с помощью Selenium	170
<i>Багатурия В. В., Сусов Р. В.</i> Преимущества и недостатки объектно-ориентированного моделирования на основе открытого стандарта UML	177
<i>Чеканова Е. Д.</i> Патентный поиск визуализации теоретико-множественного анализа сложных систем	183
РАЗДЕЛ 4. СВОБОДНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ В АДМИНИСТРИРОВАНИИ	191
<i>Борисов С.А., Наношкин А.Г., Сапегина В. С.</i> Защита корпоративной почты от спама средствами Spam Assassin	191
<i>Коновалов В.Ю.</i> Опыт сопровождения веб-приложения Redmine для автоматизации учета заявок на ремонт и обслуживание офисной техники компании	195
<i>Коршунов Э.Н., Абдрахманов Д.В.</i> Обзор утилит для администрирования вычислительных систем на платформе Linux	201
<i>Пархоменко А.С.</i> Использование свободного программного обеспечения в отделе АСУ «Транспорт» ООО «ММК-Информсервис»	207
<i>Рубан К. А.</i> Опыт внедрения свободного программного обеспечения виртуализации серверной инфраструктуры организации (на примере МГТУ им. Г.И. Носова)	213
<i>Уймин А. Г., Мешков А. Д.</i> Организация учебного процесса Уральского радиотехнического колледжа им. А. С. Попова с использованием свободного программного обеспечения	216

РОССИЙСКАЯ МОЛОДЁЖНАЯ КОНФЕРЕНЦИЯ С ЭЛЕМЕНТАМИ НАУЧНОЙ ШКОЛЫ «ТЕОРИЯ И ПРАКТИКА ПРИМЕНЕНИЯ СВОБОДНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ»

*Петеляк В.Е., Гаврилова И.В., Давлеткиреева Л.З.,
ФГБОУ ВО «Магнитогорский технический государственный
университет им. Г.И. Носова»*

Российская молодёжная конференция с элементами научной школы «Теория и практика применения свободного программного обеспечения» посвящена вопросам развития информационно-телекоммуникационных систем как приоритетного направления развития науки, технологий и техники в Российской Федерации.

Свободное программное обеспечение (СПО) в настоящее время выступает распространенной моделью бизнеса, носителем передовых технологий и экспериментальной площадкой для инноваций. Проникновение СПО в различные сферы деятельности: государственное и муниципальное управление, бизнес, индустрию разработки программного обеспечения, научные и образовательные учреждения – обуславливает необходимость привлечения внимания студентов, молодых учёных и ИТ-специалистов к проблеме применения СПО в практической деятельности. Конференция с элементами научной школы даёт возможность приобрести участникам личный практический опыт применения СПО, что позволяет расти в личном и профессиональном плане, становиться конкурентоспособными на рынке труда.

Предпосылками конференции с элементами научной школы выступили мастер-классы и семинары по вопросам использования СПО, проводимые участниками Магнитогорской Linux-группы совместно с лабораторий открытых систем университета и ИРЭ РАН с 2007 г., а также Всероссийская молодёжная конференция с элементами научной школы «Теория и практика применения свободного программного обеспечения», проходившая в октябре 2011 г.

Работа молодёжной конференции с элементами научной школы организована по четырём секциям и предполагает обсуждение применения СПО в системе среднего общего и высшего профессионального образования; актуальных вопросов в области разработки программного обеспечения, перспективных бизнес-приложений СПО, а также последних тенденций в администрировании и обеспечении информационной безопасности и защиты информации.

Законодательный переход на свободное программное обеспечение

Политика государства на внедрение свободного программного обеспечения развивается с 2008 г. и приняла более ужесточённый критерий после санкций со стороны Запада.

В марте 2008 г. была разработана Концепция развития разработки и использования свободного программного обеспечения в Российской Федерации. В документе, в частности, определены «принципы развития разработки и использования СПО». Концепция предполагала создание в период до 2010 года ряда продуктов, необходимых для государственных и образовательных учреждений. Несмотря на то, что Концепция так и не была принята, ее первоочередная роль все же сводилась к определению понятий и намерений государства в сфере разработки и внедрения СПО.

Так, в «Стратегии развития информационного общества в Российской Федерации» 2008 г. на февраль 2010 был запланирован переход общеобразовательных учреждений на использование разработанного пакета свободного программного обеспечения (ПСПО)».

В системном проекте форматирования электронного правительства в Российской Федерации, разработанном в июле 2010 г., предписывалось в типовых решениях использовать преимущественно свободно лицензируемое ПО, обеспечивающее органам власти и их поставщикам права на свободное тиражирование и модификацию. В качестве системной меры повышения эффективности бюджетных расходов на внедрение информационно-коммуникационных технологий был предусмотрен «переход на свободное программное обеспечение для реализации обеспечивающих функций и в части общесистемного программного обеспечения».

В государственной программе «Информационное общество (2011-2020 годы)» упор будет сделан на оснащение регионов не техникой, а прикладными системами, основанными на типовых решениях, которые будут использоваться в медицине, образовании и т.д

17 декабря 2010 года Председатель Правительства РФ Владимир Путин подписал распоряжение «О плане перехода федеральных органов исполнительной власти и федеральных бюджетных учреждений на использование свободного программного обеспечения (2011-2015 годы)». Председатель Правительства РФ Владимир Путин утвердил план перехода федеральных органов исполнительной власти и федеральных бюджетных учреждений на использование свободного программного обеспечения (СПО) на 2011 - 2015 годы.

В результате реализации плана было выполнено:

- в июне 2011 принят ГОСТ Р ИСО/МЭК 26300-2010 на формат OpenDocument.

- с 1 января 2012 года действует ГОСТ Р 54593-2011 «Информационные технологии. Свободное программное обеспечение. Общие положения».

- 12 марта 2014 года подписан Федеральный закон N 35-ФЗ с поправками в четвертую часть ГК РФ, регулирующих использование на территории РФ открытых лицензий.

Свободное программное обеспечение в программной инженерии

Под программной инженерией понимается дисциплина, которая связана со всеми аспектами производства программного обеспечения от начальных стадий создания спецификации до поддержки системы после сдачи в эксплуатацию. В настоящий момент классический подход к разработке сложных программных комплексов подразумевает использование единой линейки программного обеспечения: от средств моделирования до программ управления исходным кодом, тестирования и сопровождения. Как правило, такой набор могут предложить только крупные производители СУБД и средств разработки.

Несмотря на то, что программный инструментарий был одним из первых видов свободного программного обеспечения, в среде профессиональных разработчиков существует стереотип бедного функционала, плохой эргономики и низкой надёжности таких сред. Современная практика программной инженерии показывает, что это далеко не так.

Свободное программное обеспечение для создания программных комплексов воплощает в себе те же принципы, что и проприетарные среды: визуальное программирование, библиотеки классов, автоматическое журналирование и т.п. – это обусловлено тем, что программисты стремятся создать, прежде всего, удобный и полнофункциональный продукт, которым сами будут пользоваться в дальнейшем. Другими словами, для себя создавать плохое ПО никто не будет. Вторым фактором, обуславливающим качество свободных сред разработки, выступает интеграция знаний и опыта разработчиков всего мира, характерная для любых ИТ-проектов.

В последнее время использование СПО в программной инженерии становится интересно soft-компаниям ещё и потому, что отсутствие лицензионных отчислений снижает объём сметы ИТ-проекта.

Таким образом, в настоящее время актуальна проблема не столько целесообразности использования СПО в программной инженерии, сколько разработки эффективных техник его применения в производстве программного обеспечения. К сожалению, подобный опыт сложно приобрести в академической среде, необходимо быть активным участником профессиональных сообществ, знать ведущих специалистов, которые могли бы выступить своеобразными наставниками.

Использование СПО в качестве рабочего инструментария в методиках обучения программированию позволило бы упростить подготовку грамотных специалистов, готовых к ведению ИТ-проектов на базе СПО. Это положение обусловило появление второй секции Конференции - «Свободное программное обеспечение в образовании».

Свободное программное обеспечение в образовании

На текущий момент ситуация с использованием программного обеспечения в образовании сталкивается с проблемами технического и кадрового характера. Средний возраст педагогов в школах приближается к 40-45 годам, уровень владения ИКТ в опросах они сами оценивают как средний или ниже среднего. Это значит, что овладение навыками работы с программным обеспечением требует от них значительных усилий, и они не готовы постоянно их прикладывать для совершенствования опыта работы сверх необходимого.

Второй аспект связан с техническими проблемами эксплуатации устаревшей вычислительной техники в школах. Парк компьютерной техники, зачастую, включает морально устаревшие периферийные устройства, поддержку которых разработчики современных версий Linux могут не реализовать.

Рынок обучающих программных средств для школы развивается в нашей стране на протяжении последних 10-15 лет, это значит, что самые первые из них имеют жёсткие требования к операционной системе. При переходе к СПО эти программные средства не будут качественно функционировать, а покупка новых обучающих программ для бюджетной организации всегда проблематична.

Всё вышесказанное приводит к тому, что использование СПО в системе образования требует не только обновлённых методик обучения, но и грамотных специалистов, способных оказывать информационно-техническую поддержку учебного процесса. Данное положение вынесло на поверхность проблему администрирования и обеспечения информационной безопасности и защиты информации средствами СПО.

Свободное программное обеспечение в администрировании и обеспечении информационной безопасности и защиты информации

Информационная инфраструктура, построенная на базе СПО, имеет ряд положительных и отрицательных моментов. Во-первых, такая система более безопасна, поскольку компьютерных вирусов для неё нет. Во-вторых, она более защищена. Но, в то же время, построение и сопровождение такой инфраструктуры из-за её сложности требует от специалиста большой квалификации. Неточность, ошибка, недостаточная продуманность администратором своих действий пагубно отражается на работоспособности системы. Неверный выбор файловой системы может заметно снизить производительность всего аппаратно-программного

комплекса. По этой причине ценен любой опыт решения проблем администрирования систем, построенных на базе СПО.

Подбор качественного полнофункционального инструментария для сопровождения систем – вторая актуальная проблема администрирования.

Свободное программное обеспечение в бизнесе

Политика импортозамещения усложнила использование СПО в бизнесе, т.к. если ПО содержит компоненты иностранных продуктов, отвечающие за ключевой функционал, то эксперты вправе отказать во включении такого продукта в реестр. Для реального прорыва СПО-решений в нашей стране не хватает заинтересованности заказчиков, которые не спешат переходить на СПО несмотря на нестабильный курс национальной валюты.

В крупных организациях, где цена ошибки бывает очень велика, приходится учитывать все возможные риски. И на тот случай, если СПО-проект по каким-то причинам потерпит неудачу, руководство требует подстраховываться похожим проприетарным решением. Однако сделать это далеко не просто, поскольку в государственных организациях ведение двух альтернативных пилотных проектов чревато обвинениями в нецелевом расходовании средств. Кроме того, поскольку в крупных компаниях наряду с СПО широко используются и проприетарные продукты, заказчики всегда озабочены проблемами интеграции этих двух классов ПО.

Как показало анкетирование участников Конференции, интересы участников по направлениям работы распределились следующим образом (см. рисунок. 1).



Рис. 1. Распределение участников по направлениям Конференции

Для оценки результативности учебных мероприятий Конференции была разработана анкета для участников, а для слушателей подготовлены контрольные материалы в виде тестов по изученным циклам. Кроме этого был проведен конкурс презентаций кратких научных отчетов, подготовленных по результатам проведенного на семинарах обсуждения исследований, проводимых молодыми учеными. Каждый участник самостоятельно выбирал одно из направлений Конференции для выполнения отчётной работы. Все слушатели Конференции, успешно выполнившие отчётные задания, получили сертификаты о повышении квалификации.

Для публикации тезисов в сборнике материалов по итогам работы Конференции были разработаны критерии отбора научных публикаций, учитывающие соответствие тематике Конференции, инновационную направленность работы, степень реализации.

В ходе Конференции поддерживался благоприятный психологический климат среди участников. Была разработана и апробирована технология оценки качества обучения слушателей. Большое внимание отводилось неформальному общению специалистов и молодых учёных, что позволило укрепить научные связи. Иногородним докладчикам была предложена культурная программа: выезд на горнолыжный центр «Металлург – Магнитогорск», спортивное мероприятие «Полоса препятствий», просмотр Кубка Урала по маунтинбайку и др.

По итогам работы Конференции проведено анкетирование участников и анализ полученных результатов. Докладчики дали высокую оценку научному и организационно-техническому обеспечению проведения мероприятий Конференции.

Осуществлена подготовка материально-технического обеспечения Конференции: оформлена аудитория, подготовлено мультимедийное оборудование, раздаточные и презентационные материалы, организована видеосъемка и представление в свободном доступе материалов.

В ходе выполнения задания осуществлено информирование научных организаций, высших учебных заведений Российской Федерации, инновационных компаний, молодых ученых о мероприятии, условиях участия в нем и его результатах.

Для своевременного информирования участников Конференции было сделано следующее:

- обеспечено оперативное информирование участников Конференции посредством рассылки сведений о мероприятии по электронной почте;
- разработан Интернет-сайт Всероссийской молодёжной конференции с элементами научной школы «Теория и практика применения свободного программного обеспечения»;

- обеспечено оперативное информирование участников Конференции посредством публикации материалов на сайтах Конференции и Магнитогорской группы пользователей Linux;
- осуществлена публикация информации в электронных и традиционных СМИ.

Информационная поддержка мероприятия была оказана ведущими научными сайтами. К освещению мероприятий в рамках Конференции были привлечены средства массовой информации: «Диалог магнитогорцев», Магнитогорское информационное агентство «Верстов. Инфо» и др.

Видеоматериалы докладов находятся в открытом доступе на сайте по адресу [http:// http://confspo.magtu.ru/](http://http://confspo.magtu.ru/). Распределение по городам РФ иногородних участников Конференции представлено на рисунке. 2.



Рис. 2. География иногородних участников Конференции

Потребителями научного результата Конференции являются образовательные учреждения всех уровней образования, научные организации, организации и инновационные компании, работающие в сфере ИКТ.

Полученные научные результаты могут быть найдены свое применение:

- в процессе формирования тем научно-исследовательской работы студентов, аспирантов, молодых ученых;

- для получения полной и достоверной информации о текущем состоянии научных исследований по направлениям работы Конференции;
- для организации образовательного и научно-исследовательского процесса на всех уровнях системы образования;
- при планировании и проведении аналогичных мероприятий – научных молодёжных школ, конференций молодых ученых;
- в процессе организации и выполнения многосторонних проектов, написания грантов и проведения совместных исследований;
- при разработке образовательных программ, написания методической литературы для системы повышения профессиональной и научной квалификации по проблеме разработки и применения свободного программного обеспечения и др.

Возраст участников Конференции отражен на диаграмме, представленной на Рисунке. 3.



Рис. 3. Распределение участников по возрастам

Проведение Российской молодёжной конференции с элементами научной школы «Теория и практика применения свободного программного обеспечения» дало возможность эффективному освоению молодыми учеными и преподавателями наиболее результативных научно-методических отечественных достижений, стимулированию притока молодежи в сферу науки, образования и высоких технологий, увеличению числа научных и образовательных организаций, использующих передовой опыт в области свободного программного обеспечения.

РАЗДЕЛ 1. СВОБОДНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ В ОБРАЗОВАНИИ И НАУКЕ

Сюжетно-ролевое обучение работе в среде программирования Scratch

*Дегтярева К. С., Мухеева А. Р.,
ФГБОУ ВО «Магнитогорский государственный университет им.
Г.И. Носова»*

Аннотация. Статья посвящена описанию методики сюжетно-ролевого обучения в открытой среде программирования Scratch. Представлен фрагмент реализации мультипликационного фильма из набора объектов, поддерживаемых средой. Сделан вывод о целесообразности применения сюжетно-ролевого подхода для обучения.

Ключевые слова: Scratch, мультфильм, сюжетно-ролевое занятие, скрипт

Scratch – среда программирования, позволяющая создавать собственные анимированные интерактивные истории, игры и модели. В данной программе можно играть с различными объектами, видоизменять их и устанавливать различные формы взаимодействия между ними [2].

Исходя из этого, для изучения программирования в данной среде лучше использовать сюжетно-ролевой подход, а не традиционный, в соответствии с которым сначала изучается синтаксис и команды языка[3].

Сюжетно-ролевая игра формируется в процессе жизни ребенка, в ходе его общения с окружающими людьми. Как деятельность сюжетно-ролевая игра представляет собой сложное образование, состоящее из следующих компонентов: целевого, потребностно-мотивационного, содержательного, операционного, результативного [1].

На наш взгляд сюжетно-ролевое обучение можно рассматривать как один из наилучших способов преподавания и контроля усвоения тяжелых для детского восприятия тем [3-15]. Рассмотрим наше предположение на проведении сюжетно-ролевого занятия.

Целью данного занятия является реализация сюжетно-ролевого подхода с использованием среды Scratch на примере проведения внеурочного занятия по теме: «Правила безопасного использования сети Интернет».

Для того, чтобы мотивировать учащихся на изучение темы им предлагается создать мультфильм, в котором предполагается следующий сюжет.

Действующие лица в мультфильме выступают в виде 5 скриптов: Laptop (Ноутбук), Cat2 (Кошка), Alex (Девочка Алекс), Ghoul (вирус Nimda), Wizard2 (антивирус NOD 32) (рисунок 1).

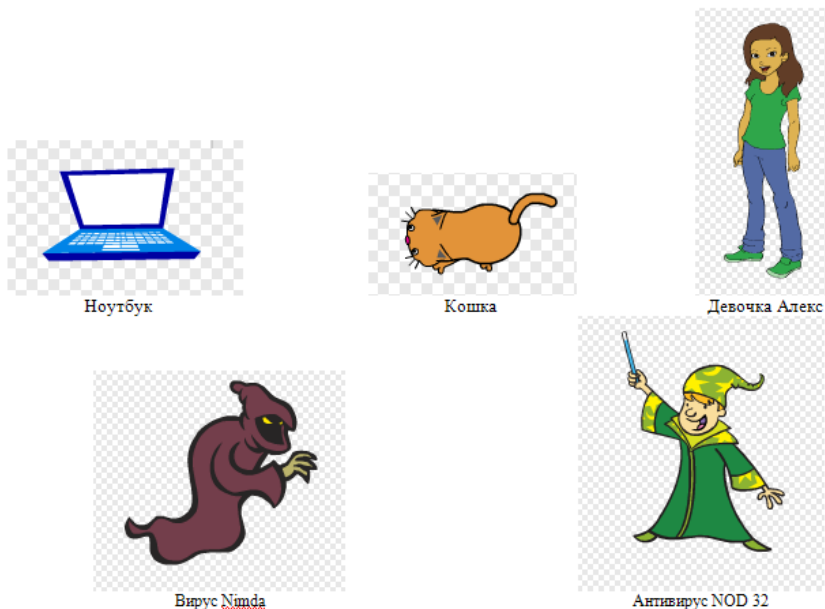


Рис. 1. Герои мультфильма

Действия героев происходят в комнате девочки Алекс. Будучи не опытным пользователем ПК, Алекс поймала вирус Nimda на одном из сайтов в Интернете, что повлекло за собой «смерть» её компьютера. Положение спасает кошка девочки, призвавшая на помощь антивирус NOD 32, который победил злой вирус и вернул к жизни любимый ноутбук Алекс. Действия заканчиваются тем, что девочка подводит итоги, извлекая из произошедшего урок.

Опишем в среде Scratch фрагмент реализации данного мультфильма. Рассмотрим спрайт laptop (ноутбук) (рисунок 2). В нашем примере laptop выполняет следующие команды: смена костюма, разговор и ожидание. Так же для выполнения его действий мы обращались к таким скриптам как «говорить», «передать», «спрятаться».

Для того, чтобы ноутбук начал выполнять выбранные нами действия, нужно воспользоваться специальной командой (связанной с кнопкой контроль). На ней написано «когда щелкнут по» и изображен зеленый флажок. Если взять эту команду, поместить в поле скриптов и соединить с командами «сменить костюм» и «показаться», то

программа станет запускаться при нажатии на кнопку запуска, а сам герой приобретет темный экран и появится на экране (рисунок 3).



Рис. 2. Laptop (ноутбук)

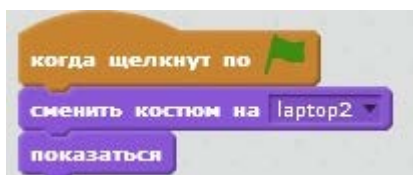


Рис. 3. Действия

Далее в анимации появляются другие действующие персонажи, а так как наш ноутбук должен оставаться в одном и том же положении, то мы просто разрываем скрипты. После разрыва герой получает сообщение от другого персонажа. Это необходимо для того чтобы начало выполнений действий ноутбука не накладывались по времени на действия других, а следовали друг за другом. В этом нам поможет скрипт «Когда я получу...». Данный скрипт соединяем с теми командами, которые должен выполнить герой «ожидание», «говорить ... в течение ... секунд» и снова «сменить костюм» (рисунок 4).

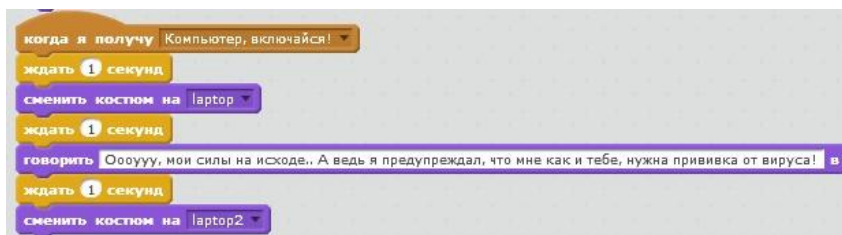


Рис. 4. Действия

Так же действия героя могут начинаться после смены фона. Задействован скрипт «когда фон меняется на ...» (рисунок 5).

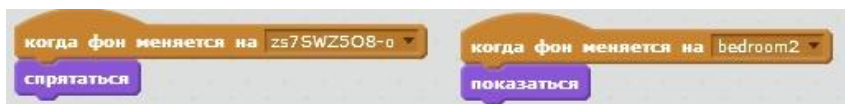


Рис. 5. Смена фона

Затем, следуя сценарию и описанным выше этапам его реализации, завершаем работу с героем (рисунок 6).

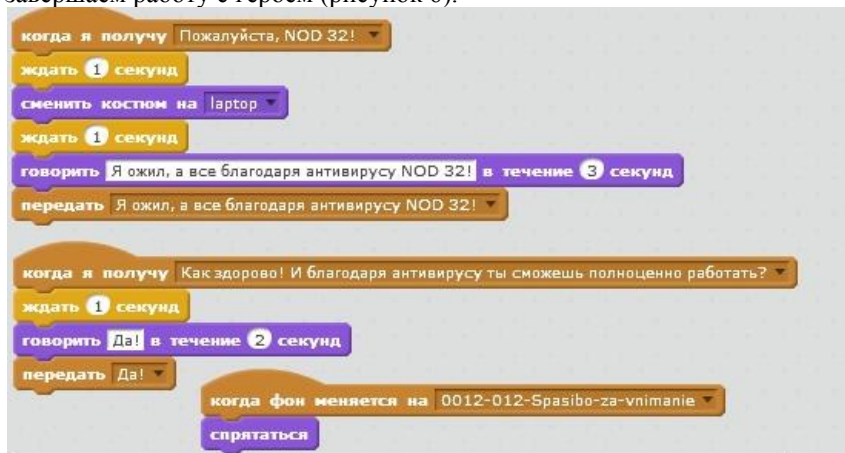


Рис. 6. Действия

Подводя итоги, можно сказать, что сюжетно-ролевой подход можно использовать не только при обучении работы в Scratch, но и для изучения детьми тем любого направления. Эффективность освоения материала будет выше, что во многом облегчит задачу любого преподавателя.

Список использованных источников

1. Обучение сюжетно-ролевой игре дошкольников с проблемами в интеллектуальном развитии: учеб.-метод. пособие / Под ред. Л.Б. Баряевой, А.П. Зарин, Н.Д. Соколовой. - СПб: ЛОИУУ, 1996. -95 с.
2. Scratch. Общедоступное программирование в Scratch. - Режим доступа: <http://scratch.uvk6.info/>
3. Зеркина Е.В., Литвин А.В. Преподавание анимационной среды программирования СКРЕТЧ в системе дополнительного образования (на примере кружка) // Актуальные проблемы прикладной информатики и методики обучения информатике : материалы молодеж. всеросс. науч.-практ. конференции / отв. ред. И.Н. Слинкина. – Шадринск : ШГПИ, 2009. – 265 с. – с. 66-68.

4. Курзаева Л.В., Чусавитина Г.Н. Подготовка будущих педагогических кадров к превенции киберэкстремизма среди молодежи: моделирование процесса установления требований к процессу профессиональной подготовки // *Фундаментальные исследования*. -2014. -№ 12-5. -С. 1078-1082.

5. Курзаева Л.В. Развитие информационно-аналитической компетенции будущего IT-специалиста с использованием компьютерного имитационного моделирования профессиональных задач // *Научные труды SWorld*. 2007. Т. 14. № 3. С. 16а-18.

6. Курзаева Л.В., Чусавитина Г.Н. К вопросу о формировании требований к компетенциям личности в области информационной безопасности в системе высшего профессионального образования // *Фундаментальные исследования*. 2013. № 8 (часть 5). С. 1203-1207; Режим доступа: www.rae.ru/fs/?section=content&op=show_article&article_id=10001321

7. Курзаева Л.В. Ситуационные тесты в системе оценки результатов обучения личности (на примере IT-направлений подготовки) /Л.В. Курзаева // *Научные труды SWorld*. - 2013. - Т. 16. - № 3.- С. 61-63.

8. Курзаева Л.В. Тестовая оценка результатов обучения личности в парадигме компетентностного подхода (на примере IT- направлений подготовки) // *Современные информационные технологии и IT-образование*. - 2013. - № 9. - С. 151-156.

9. Лактионова Ю.С. Информатика : учебное пособие / Ю. С. Лактионова ; М-во образования и науки Российской Федерации, Федеральное гос. образовательное учреждение высш. проф. образования ГОУ ВПО "Магнитогорский гос. ун-т". Магнитогорск, 2010. – 180 с.

10. Лактионова Ю.С. Модель развивающего обучения учащихся старших классов средней общеобразовательной школы в процессе изучения «Информатики и ИКТ» / *Сибирский педагогический журнал*. 2010. № 7. С. 278-284.

11. Лактионова, Ю.С. Активизация учебной деятельности учащихся посредством дидактических игр//*Информатика и образование*.-Москва, 2010.-№5.-с.65-67.

12. Лактионова Ю.С. Структурные элементы дидактической системы развивающего обучения информатике // *Научные труды SWorld*. 2009. Т. 18. № 4. С. 33-35.

13. Лактионова Ю.С., Брябрина Л.С. Использование игровых технологий в процессе обучения школьников информатике // *Научные труды SWorld*. 2013. Т. 16. № 3. С. 43-46.

14. Чернова Е.В. Современные сетевые сервисы в обучении основам информационной безопасности // *Новые информационные технологии в образовании : материалы Всероссийской заочной*

электронной науч.-практ. конференции, 15-16 нояб. 2012 г. / Сев.-Вост. гос. ун-т ; [отв. ред.Т.А. Брачун]. – Магадан : СВГУ, 2013. – 128 с. : ил. – с. 79-80.

15. Чусавитина Г.Н. Подготовка будущих учителей к обеспечению информационной безопасности: монография /Г.Н. Чусавитина, Л.З. Давлеткиреева, Л. В. Курзаева, М.О. Чусавитин. -Магнитогорск: МаГУ, 2013. -188 с.

Опыт использования свободного программного обеспечения при подготовке ИТ - специалистов

*Гусева Т.Ф.,
ФГБОУ ВО «Магнитогорский государственный технический
университет им. Г.И. Носова»*

Аннотация. В работе рассматривается проблема выбора программного обеспечения (ПО) при подготовке ИТ-специалистов в различных сферах деятельности. Также приведены примеры программных средств, которые являются хорошей альтернативой проприетарному программному обеспечению.

Ключевые слова программное обеспечение (ПО), свободное программное обеспечение (СПО), проприетарное ПО, средства проектирования.

Быстрое развитие компьютерной техники и программного обеспечения (ПО) в последние годы привело к тому, что компьютерные технологии проникли во все сферы человеческой деятельности [1]. На сегодняшний день невозможно представить себе высококвалифицированного специалиста в любой отрасли, не владеющего ИТ-технологиями. Вследствие этого у преподавателей вузов встает вопрос выбора программного обеспечения для обучения студентов, в том числе при подготовке ИТ-специалистов [2,7]. Традиционно во многих вузах в учебном процессе используются проприетарные (несвободные) продукты различных компаний, что приводит к возникновению проблем. Для закрепления материала и выполнения домашнего задания студенту необходимо дома установить такое же программное обеспечение, как в компьютерных классах университета, но многие студенты не могут позволить себе проприетарное ПО. Решением данной проблемы является переход на свободное программное обеспечение (СПО), которое пользователь может улучшать и запускать свою улучшенную версию, изучать принципы

работы программы, копировать, распространять и использовать в своих целях [3].

У свободного ПО есть свои преимущества перед проприетарным ПО:

1. Отсутствие затрат на приобретение лицензионного программного обеспечения. Для установки ПО на всех компьютерах образовательной организации (ОО) достаточно купить одну коробочную версию СПО, которая стоит порядка 3500 руб., либо приобрести бесплатно, скачав программу из Интернета.

2. Имеется возможность доработать программу под конкретные нужды пользователя.

3. Подключение компьютеров с установленным СПО в локальную сеть ОО и их настройка осуществляются достаточно легко.

4. Компьютеры со свободным программным обеспечением меньше подвержены атакам компьютерных вирусов.

5. Используя СПО, клиент не зависит от разработчиков программных продуктов и поставщиков услуг.

6. Открытый исходный код свободного ПО, как правило, качественнее, чем закрытый, так как закрытый исходный код не отображает ошибки или недочеты разработчика [4].

Всё это позволяет рекомендовать использовать именно свободные программы в университетах в учебном процессе и научных исследованиях. Приведем примеры свободных прикладных программ, применяемых в образовательном процессе:

1. StarUML - программный инструмент моделирования, который поддерживает UML (Унифицированный язык моделирования). Целью создания программы было стремление заменить коммерческие версии средств UML, например, Rational Rose и др. На рисунке 1 представлен интерфейс StarUML.

Преимущества:

- позволяет производить профессиональные результаты моделирования;

- позволяет создавать плагины;

- распространяется совершенно бесплатно.

Недостатки:

- сложная программа для начинающих;

- плагины требуют знаний в области программирования.

2. Среда разработки Eclipse является открытой и свободной, скачать ее можно с официального сайта Eclipse.org. Также на официальном сайте доступен широкий выбор IDE, заранее настроенный для работы с тем или иным языком программирования. Платформа расширяется с помощью плагинов, каждый из которых может дополнять

поведение Eclipse. Например, добавляя новые редакторы, пункты меню и т.д. На рисунке 2 представлен интерфейс среды разработки Eclipse.

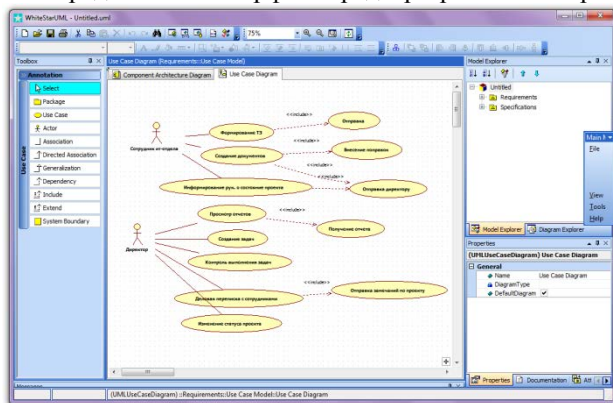


Рис. 1. Интерфейс StarUML

Основными особенностями Eclipse являются:

- кроссплатформенность;
- программирование на множестве языков;
- для разработки иных инструментов среда является фреймворком и предлагает широкую подборку интерфейсов прикладного программирования для создания модулей;
- создание клиентского ПО с помощью подхода Rich Client Platform.

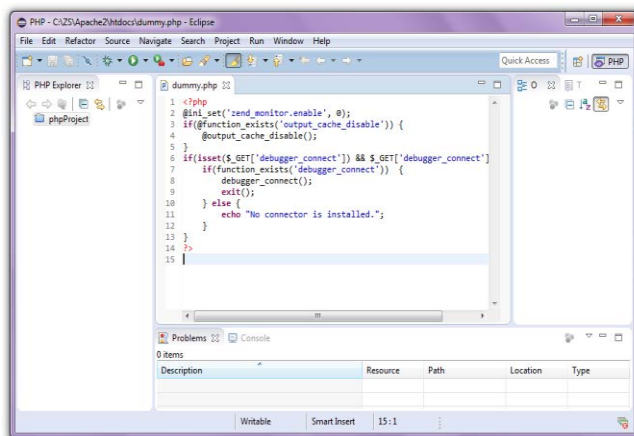


Рис. 2. Интерфейс Eclipse

3. FreePascal (freepascalcompiler) - профессиональный компилятор языка Pascal, распространяемый под лицензией (L) GPL и работающий под Windows, Linux, MacOSX, FreeBSD, и не только. На рисунке 3 представлен интерфейс FreePascal.

Основные возможности Free Pascal:

- кроссплатформенность;
- широкие возможности – всевозможные расширения и макросы, некоторые функции на уровне Delphi;
- быстрая компиляция – по сравнению с другими, на 20%.

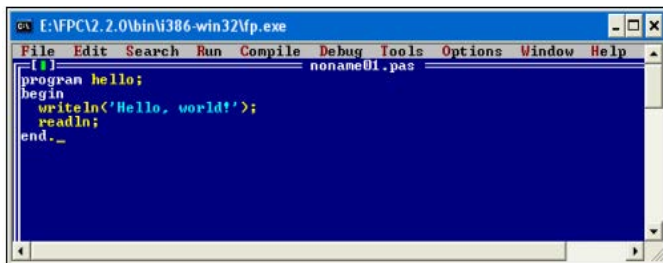


Рис. 3. Интерфейс FreePascal

4. Apache Open Office — свободный пакет офисных приложений. Отлично заменяет коммерческую версию пакета Microsoft Office. Так как на Linux системах нет MS Office, то программистами для него был написан свой собственный пакет. Сейчас Open Office официально поддерживается на платформах Linux, Microsoft Windows, MacOSXIntel. На рисунке 4 представлен интерфейс Apache Open Office.



Рис. 4. Apache Open Office

Основным преимуществом OpenOffice является то, что она доступна русская версия, которая работает на любом персональном

компьютере. Кроме того в программе реализована возможность открывать документ на том месте, на котором были внесены последние изменения.

5. CA ERwin Data Modeler Community Edition - это простой инструмент моделирования данных, который включает в себя подмножество функций флагманского продукта CA ERwin Data Modeler Standard Edition. Программа является отличным средством получения навыков моделирования, как у студентов, так и у новичков[4,5]. На рисунке 5 представлен интерфейс CA ERwin Data Modeler Community Edition r9.5.

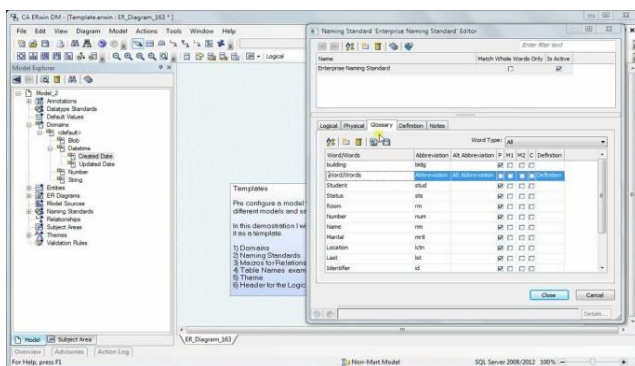


Рис. 5. Интерфейс CA Erwin Data Modeler Community Edition r9.5

Эти и другие программы являются хорошей альтернативой коммерческих пакетов прикладных программ для работы с графической, звуковой и видеоинформацией.

Свободное программное обеспечение позволяет осуществлять более качественную подготовку специалистов по информационным технологиям, но полностью переходить на использование СПО в настоящее время нецелесообразно, поскольку после окончания высших учебных заведений большинству выпускников на своих рабочих местах необходимо будет сначала осваивать, а затем использовать коммерческое программное обеспечение.

Список использованных источников

1. Масленникова О.Е., Назарова О.Б. Роль и место проектной работы студентов в их профессиональном становлении / О.Е.Масленникова, О.Б.Назарова//Новые информационные технологии в образовании. Материалы VIII Международной научно-практической конференции. Российский государственный профессионально-педагогический университет. 2015. – С. 470-474.

2. Назарова О.Б., Масленникова О.Е., Новикова Т.Б., Давлеткиреева Л.З. Преемственность дисциплин кафедры по направлению «Прикладная информатика» и межпредметная координация как фактор повышения эффективности образовательного процесса / О.Б. Назарова, О.Е. Масленникова, Т.Б. Новикова, Л.З. Давлеткиреева // Актуальные проблемы современной науки, техники и образования. 2015. - Т. 2. - № 1. - С. 145-148.

3. Назарова О.Б. Реализация принципа преемственности в построении учебных курсов специальности Прикладная информатика (в экономике) на основе Case-технологий / О.Б. Назарова // Фундаментальные исследования, -2007.- № 6. -С. 46.

4. Давлеткиреева Л.З., Махмутова М.В. Инновационная модель подготовки ИТ-специалиста в образовательной среде вуза/Л.З. Давлеткиреева, М.В. Махмутова//Современные информационные технологии и ИТ-образование. -2012. -№ 8. -С. 118-129.

5. Давлеткиреева Л.З., Скокова И.К. Обоснование инструментальных средств моделирования бизнес-процессов проведения мероприятий с использованием дистанционных технологий/Л.З. Давлеткиреева, И.К. Скокова//Современные технологии и управление: сб. научных трудов III Международной научно-практической конференции. - Светлый Яр.: Филиал ФГБОУ ВПО Московский государственный университет технологий и управления имени К. Г. Разумовского, 2014. -. - С. 694-697.

6. Давлеткиреева Л.З., Назаров В.О., Мусыгина А.А. Разработка технологической модели обработки данных и компонентов сети для КИС//Современные инновации в науке и технике : сб. научных трудов 4-ой Международной научно-практической конференции (17 апреля 2014 года)/редкол.: Горохов А.А. (отв. Ред.); В 4-х томах, Том 2., Юго-Зап. гос. ун-т. Курск, 2014. 413 с.-С.17-21

7. Давлеткиреева, Л. З. Информационно-предметная среда как средство профессиональной подготовки будущих специалистов в университете: автореф. дис.. канд. пед. наук/Л. З. Давлеткиреева. Магнитогорск, 2006. -23 с.

Использование Scratch для разработки интерактивных дидактических средств

*Емельянова Ю. А., Сироткина А.С.,
ФГБОУ ВО «Магнитогорский государственный университет им.
Г.И. Носова»*

Аннотация. В статье приводится пример разработки обучающих анимационных фильмов в среде Scratch. Подтверждается

целесообразность применения данного инструментария для разработки интерактивных дидактических средств.

Ключевые слова: Scratch, дидактические средства, спрайты, диалог.

Scratch (Скретч) – визуальная объектно-ориентированная среда программирования для обучения школьников младших и средних классов [2].

В 2003 году группа исследователей под руководством Митчела Резника из MITMediaLab решила сделать общедоступный язык программирования, а через 4 года появился Скретч. Делать в нём компьютерные программы могли даже школьники [1]. Рассмотрим применение данной среды по иному назначению – разработки интерактивного дидактического средства.

Целью данного дидактического средства является обучение и контроль освоения темы: «Правила безопасного использования сети Интернет». Этапы разработки данного средства включают:

1. Разработка сценария мультфильма.
2. Разработка тестовой части.
3. Создание мультфильма.

1. Разработка сценария мультфильма.
Сценарий мультфильма следующий.

Персонажи: мама, дочка, обезьянка.

Первая комната: Прихожая.

Мама заходит в комнату и ищет свою дочку Ксюшу.

▪ Мама: Ксюша, ты где? (Думает) Где она может быть? Может быть в гостиной?

Вторая комната: Гостиная.

Мама ищет дочку в гостиной.

▪ (Думает): Хмм... Её тут нет. Значит она в своей комнате.

Третья комната: Спальня Ксюши.

Мама и дочка общаются между собой и обсуждают проблему, которая образовалась.

▪ Мама: Доча, вот ты где? Почему ты не отзываешься?

▪ Дочь: Мааааам, кажется, я сломала компьютер!! Я зашла в интернете на сайт, а компьютер выключился и не включается! Что же мне теперь делать?

▪ Мама: Ксюша, надо было думать, прежде чем заходить на этот сайт! Давай я тебе расскажу, как правильно нужно пользоваться интернетом! Правило № 1. Никогда не заходи на не проверенные сайты. Поняла?

▪ Дочь: Больше не буду, обещаю!

▪ Мама: Правило № 2. Не скачивай и не открывай неизвестные файлы.

- Дочь: Хорошо, мамочка.
- Мама: Правило № 3. Никогда не рассказывай о себе незнакомым людям: где ты живешь, учишься, свой номер телефона!

- Дочь: Я поняла.

- Мама: Правило № 4. Никогда не открывай подозрительные сообщения или вложения электронной почты, полученные от незнакомых людей. Правило № 5. Если тебя кто-то расстроил или обидел в Интернете, расскажи всё мне или папе. Мы обязательно тебе поможем.

- Дочь: Хорошо, я обязательно вам расскажу.

- Мама: А теперь давай проверим, как ты запомнила эти простые правила.

Тест (Дидактические аспекты разработаны на основе рекомендаций [2-12]).

- Обезьянка: Привет, ребята! Давайте поможем Ксюше решить мамин тест. Правила очень простые. Если ты согласен, нажми цифру "1". А если не согласен, нажми цифру "2".

- Обезьянка: Спасибо, что ты помог Ксюше пройти этот тест! Надеюсь, что ты запомнил простые правила безопасного использования Интернета!

Третья комната: Спальня Ксюши.

Ксюша благодарит маму и ребят, которые помогли пройти ей тест.

- Дочь: Спасибо, мамочка, за то, что рассказала эти правила!

- Мама: Не за что, доченька! Всегда рада тебе помочь!

2. Разработка тестовой части.

Для разработки тестовой части были определены контрольные вопросы:

1. Можно ли переходить по не проверенным ссылкам? (Если нажмут 1, то ответит: «Жаль, но это неправильный ответ», если нажмёт 2, то ответ: «Молодец! Это правильный ответ!»).

2. Можно ли посторонним людям рассказывать о себе личную информацию? (Если нажмут 1, то ответит: «Жаль, но это неправильный ответ», если нажмёт 2, то ответ: «Молодец! Это правильный ответ!»).

3. Можно ли скачивать игры с проверенных сайтов? (Если нажмут 2, то ответит: «Жаль, но это неправильный ответ», если нажмёт 1, то ответ: «Молодец! Это правильный ответ!»).

4. Будешь ли ты открывать подозрительные сообщения от незнакомцев? (Если нажмут 1, то ответит: «Жаль, но это неправильный ответ», если нажмёт 2, то ответ: «Молодец! Это правильный ответ!»).

5. Расскажешь ли ты родителям, если кто-то обидел тебя в Интернете? (Если нажмут 2, то ответит: «Жаль, но это неправильный ответ», если нажмёт 1, то ответ: «Молодец! Это правильный ответ!»).

1. Создание мультфильма.

В нашем мультфильме задействовано 3 спрайта и 5 фонов.

Для того, чтобы создать диалог в данной программе, необходимо собрать монологи для двух спрайтов, с помощью скриптов. В нашем мультфильме диалог происходит между Дочерью и Матерью, вот пример того, как выглядят скрипты, создающие мультфильм (рисунок 2).



Спрайт – Дочь (Ксюша)



Спрайт – Мама



Спрайт - Помощник

Рис. 1. Спрайты

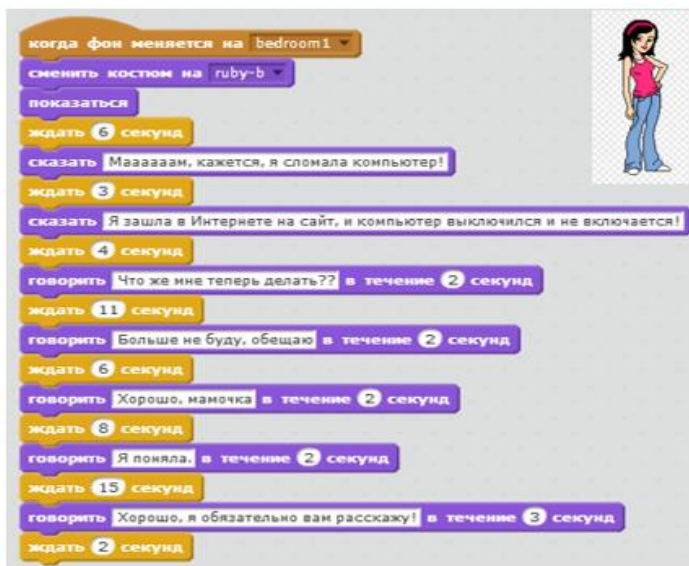


Рис. 2. Диалог дочери в комнате

Так как в нашем мультфильме присутствует тестовая часть, мы собрали скрипты объяснения правил и самого теста (рисунок 4).

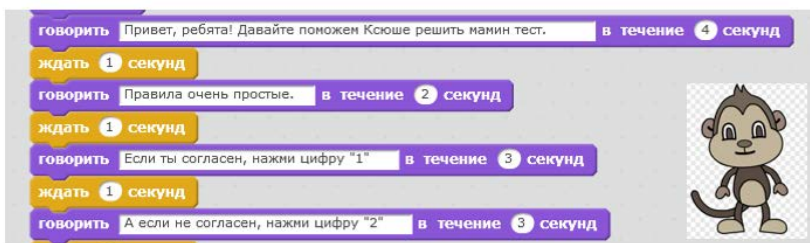


Рис. 3. Правила

Скрипт предоставления одного из вопросов и обработки ответов, представлен на рисунке 5:

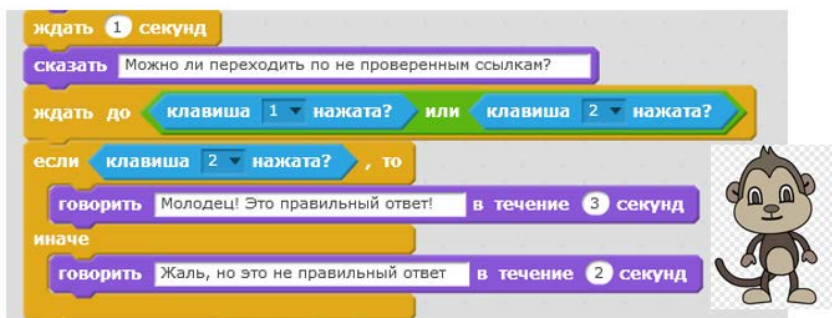


Рис. 5. Тест

Таким образом, мы показали, как можно использовать Скретч в образовательных целях. Преимуществами данного способа являются интерактивность, красочность, наглядность. Такой метод обучения очень понравится детям, даже пассивные ученики не останутся в стороне.

Список использованных источников

1. Scratch. Общедоступное программирование в Scratch. - Режим доступа: <http://scratch.uvk6.info/>
2. Зеркина Е.В., Литвин А.В. Преподавание анимационной среды программирования СКРЕТЧ в системе дополнительного образования (на примере кружка) // Актуальные проблемы прикладной информатики и методики обучения информатике : материалы молодеж. всеросс. науч.-практ. конференции. – Шадринск : ШГПИ, 2009. – 265 с. – с. 66-68.
3. Курзаева Л.В., Чусавитина Г.Н. Подготовка будущих педагогических кадров к превенции киберэкстремизма среди молодежи:

моделирование процесса установления требований к процессу профессиональной подготовки // *Фундаментальные исследования*. -2014. -№ 12-5. -С. 1078-1082.

4. Курзаева Л.В., Чусавитина Г.Н. К вопросу о формировании требований к компетенциям личности в области информационной безопасности в системе высшего профессионального образования // *Фундаментальные исследования*. 2013. № 8 (часть 5). С. 1203-1207; Режим доступа: www.rae.ru/fs/?section=content&op=show_article&article_id=10001321

5. Курзаева Л.В. Ситуационные тесты в системе оценки результатов обучения личности (на примере ИТ-направлений подготовки) /Л.В. Курзаева // *Научные труды SWorld*. - 2013. - Т. 16. - № 3.- С. 61-63.

6. Курзаева Л.В. Тестовая оценка результатов обучения личности в парадигме компетентностного подхода (на примере ИТ- направлений подготовки) // *Современные информационные технологии и ИТ-образование*. - 2013. - № 9. - С. 151-156.

7. Лактионова Ю.С. Информатика : учебное пособие / Ю. С. Лактионова ; М-во образования и науки Российской Федерации, Федеральное гос. образовательное учреждение высш. проф. образования ГОУ ВПО "Магнитогорский гос. ун-т". Магнитогорск, 2010. – 180 с.

8. Лактионова Ю.С. Модель развивающего обучения учащихся старших классов средней общеобразовательной школы в процессе изучения «Информатики и ИКТ» / *Сибирский педагогический журнал*. 2010. № 7. С. 278-284.

9. Лактионова, Ю.С. Активизация учебной деятельности учащихся посредством дидактических игр//*Информатика и образование*.-Москва, 2010.-№5.-с.65-67.

10. Ращупкин А.А., Чернова Е.В. Метод проектов в подготовке школьников к защите от киберпреступлений // *Материалы Всероссийской научно-практической конференции «Современные проблемы информационных систем и информационных технологий»*. – Кизляр, 2013. – 322с. – с. 212-218.

11. Чернова Е.В. Современные сетевые сервисы в обучении основам информационной безопасности // *Новые информационные технологии в образовании : материалы Всероссийской заочной электронной науч.-практ. конференции*, 15-16 нояб. 2012 г. / Сев.-Вост. гос. ун-т ; [отв. ред.Т.А. Брачун]. – Магадан : СВГУ, 2013. – 128 с. : ил. – с. 79-80.

12. Чусавитина Г.Н. Подготовка будущих учителей к обеспечению информационной безопасности: монография /Г.Н. Чусавитина, Л.З. Давлеткиреева, Л. В. Курзаева, М.О. Чусавитин. -Магнитогорск: МаГУ, 2013. -188 с

Реализация системы автоматической оптимизации в Scilab

*Казакова Т. В.,
ФГБОУ ВО «Магнитогорский государственный университет им.
Г.И. Носова»*

Аннотация. В работе представлено визуальное моделирование и организация алгоритмов управления в среде XCos пакета Scilab. Рассмотрен иллюстративный пример решения задачи САО при использовании способа поиска экстремума по запоминанию скорости изменения выходного параметра. Обозначен набор инструментальных средств для реализации имитационной модели технологического процесса.

Ключевые слова: имитационная модель, система автоматической оптимизации, экстремальное регулирование, среда Xcos, пакет SciLab, алгоритм автоматической оптимизации.

Пакет Scilab предназначен для выполнения научных и инженерных вычислений. По своим возможностям пакет сопоставим с известным математическим пакетом Mathcad, а по своему интерфейсу похож на пакет Matlab. Но при этом пакет прикладных программ Scilab находится в свободном распространении. Существуют версии для различных операционных систем: ОС Linux, семейство Windows и MacOS.

Scilab включает в себя развитый язык программирования и большую библиотеку численных алгоритмов. Он подходит для сложных математических расчетов при научно-исследовательских работах.

Данный пакет содержит в себе XCos, который удобен и понятен для специалистов, работающих с оборудованием. Он представляет набор инструментов, который добавляет графическую разработку систем управления.

Обзор методов реализации алгоритмов. Решение задач синтеза и реализация систем управления требуют выполнения предварительного моделирования с последующей коррекцией алгоритмов на технологическом объекте для исключения ошибок, которые могут стать причиной повреждения дорогостоящего оборудования. Язык Scilab является интерпретируемым языком высокого уровня, что позволяет манипулировать математическими конструкциями, будь то матрицы или полиномы, напрямую. Это помогает быстро и просто составлять блоки программы. Пользователи данного пакета могут разрабатывать собственные модули расширения, что бы решать конкретные задачи. Scilab предоставляет чрезвычайно богатый набор средств для научных и инженерных расчетов. Хотя первоначальный акцент при разработке

пакета был сделан на матричную алгебру, вскоре функциональные возможности расширились настолько, что охватили большинство разделов научных вычислений [1].

Проанализировав состав и возможности пакета математических инструментов Scilab, можно заметить, что он является альтернативой популярной среды Matlab. Так в чем же отличия и преимущества?

В первую очередь это, конечно, бесплатность. Среда Scilab имеет лицензию CeCILL, т.е. является свободным программным обеспечением с открытым исходным кодом.

Вторым фактором, полезным для инженеров, которым в силу специфики работы требуется мобильность, является малый объем инструментария. Scilab легко запускается с USB-носителя и занимает очень мало места. Аналогичные программные продукты такой возможностью не обладают.

Следующим преимуществом является возможность запуска программы непосредственно из терминала компьютера, что позволяет производить автоматизированные вычисления.

Стоит так же отметить кроссплатформенность данной среды. В отличии от Matlab вычисления можно производить не только из под ОС Windows, но и из под любой UNIX-системы. Конечно, у данной среды есть и недостатки. К ним можно отнести использование прежде всего численных подходов для вычисления. Этот недостаток может сказаться на точности расчетов.

Постановка задачи и процедура создания имитационной модели в среде Scilab. Для систем автоматической оптимизации требуется апробировать технологию имитационного моделирования и тестирования в среде XCos пакета Scilab.

Рассмотрим объект управления с унимодальной статической характеристикой, тогда задачей САО является нахождение и поддержание минимума или максимума величины процесса. Для большинства производственных процессов могут быть определены такие экстремальные зависимости и, соответственно, использованы САО для поддержания характеристик процесса на оптимальном значении. Так, в работе [2] рассматриваются использование САО для процессов сжигания топлива, в работах [3-5] – процессы оптимизации электрических и энергетических режимов в дуговых сталеплавильных печах и агрегатах доводки стали, в работах [6,7] – процессы внепечной обработки стали.

Имитационное моделирование САО

Работоспособность САО при использовании способа поиска экстремума по запоминанию скорости изменения выходного параметра и остановкой исполнительного механизма в момент достижения максимальной скорости выходного оптимизируемого параметра и эффективность экстремального регулятора можно проверить на

имитационной модели реализованной в программной среде Xcos пакета SciLab.

Структура системы управления в среде XCos пакета SciLab представлена на рисунке 1.

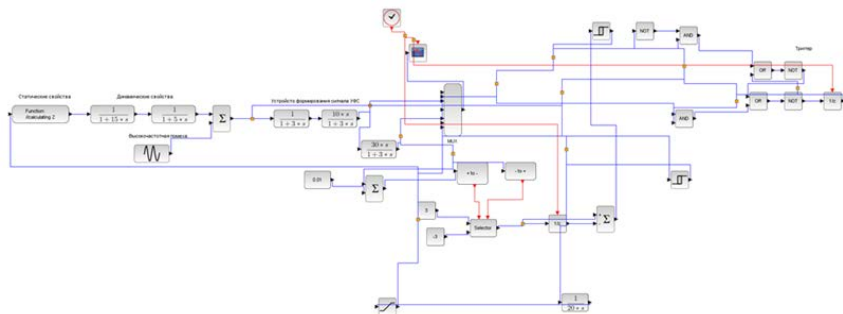


Рис. 1. Структура САУ в среде Xcos

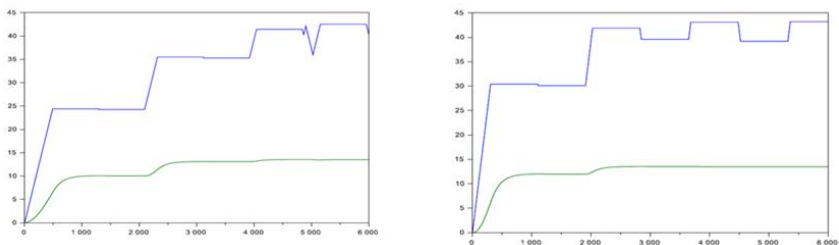


Рис. 2. Переходные процессы в системе при разных параметрах настройки

В приведенной модели учитываются инерционные свойства объекта управления и также введена дополнительная помеха для имитации работы объекта в реальных условиях.

Для реализации динамических параметров объекта в среде Xcos элемент задержки аппроксимируется цепочкой из последовательно включенных апериодических звеньев первого порядка, и на этапе моделирования можно выполнить необходимые численные эксперименты, подтверждающие работоспособность и эффективность алгоритмов управления САУ.

Результаты имитационного моделирования и тестирования экспериментальной системы, представленные (рисунке 2) в виде переходных процессов, отражают работоспособность имитационной модели САУ при использовании способа поиска экстремума по запоминанию скорости изменения выходного параметра.

Выводы

Все результаты расчетов были в пределах допустимой погрешности. Scilab является бесплатной средой разработки с открытым исходным кодом и высоким качеством проработки и является очень быстрой системой прототипирования. Система может быть использована широким спектре технических наук, как альтернатива Matlab/ Simulink. Система также может быть применена в качестве среды разработки систем управления. Так как свободный софт сейчас становится все более и более популярен, следует, что навыки разработки приложений в данной среде не просто удобны, а необходимы.

Список использованных источников

1. Леонтьев Н.А. Применение пакета математических программ Scilab для моделирования обработки нелинейных оптических сигналов // Современные научные исследования и инновации. 2015. № 5 [Электронный ресурс]. Режим доступа: <http://web.snauka.ru/issues/2015/05/51975> (дата обращения: 31.03.2016).
2. Андреев, С.М. Оптимизация режимов управления нагревом заготовок в печах проходного типа [Текст]: монография. / С.М. Андреев, Б.Н. Парсункин/ - Магнитогорск: Изд-во Магнитогорск. гос. тех. ун-та им. Г.И. Носова, 2013. - 376 с.
3. Парсункин, Б. Н. Синтез системы оптимального управления электрическим режимом сверхмощной дуговой сталеплавильной печи ДСП-180 [Текст] /Б.Н. Парсункин, С.М. Андреев, Е.Н. Ишметьев, М.В. Усачев, Е.С. Михальченко, А.К. Наливкин // Мехатроника, автоматизация, управление. - 2010. -№8. -С. 11-18.
4. Ишметьев, Е.Н. Оптимизация энергетического режима работы электродуговой печи [Текст] /Е.Н. Ишметьев, С.М. Андреев, Б.Н. Парсункин, З.Г. Салихов, М.В. Усачев, М.Ю. Рябчиков// Известия Высших учебных заведений. Черная металлургия. 2007. -№5. –С.23-27.
5. Парсункин, Б. Н. Автоматизация и оптимизация управления выплавкой стали в электродуговых печах [Текст] / Б.Н. Парсункин, С.М. Андреев, О.С. Логунова/ – Магнитогорск: Изд-во Магнитогорск. гос. техн. ун-та им. Г.И. Носова, 2012. -304 с.
6. Ишметьев, Е.Н. Автоматизация и оптимизация управления технологическими процессами внепечной доводки стали [Текст] / Е.Н. Ишметьев, С.М. Андреев, Б.Н. Парсункин, З.Г. Салихов, У.Б. Ахметов/ – Магнитогорск: Изд-во Магнитогорск. гос. техн. ун-та им. Г.И. Носова, 2008. -311 с.
7. Парсункин, Б.Н. Оптимизация работы установки циркуляционного вакуумирования стали [Текст]/ Б.Н. Парсункин, С.М.

Андреев // Вестник Магнитогорского гос. техн. ун-та им. Г.И. Носова. - 2010. - №1. - С.23-28.

8. Skiba Grzegorz, Żabiński Tomasz, Bożek Andrzej Rapid Control Prototyping with Scilab/Scicos/RTAI for PC-based and ARM-based Platforms [Текст]/ Skiba Grzegorz, Żabiński Tomasz, Bożek Andrzej //PROCEEDINGS OF THE IMCSIT Rzeszów University of Technology. -2008. -№3. –С.1-6.

Свободное программное обеспечение в электронном обучении

*Карманова Е.В., доцент, к. п. н.,
ФГБОУ ВО «Магнитогорский государственный технический
университет им. Г.И. Носова»*

Аннотация. В статье представлен обзор существующего свободного программного обеспечения для реализации электронного обучения. Дана классификация соответствующего программного обеспечения. Также обоснована целесообразность использования свободного программного обеспечения в образовании.

Ключевые слова: электронное обучение, свободное программное обеспечение, бесплатное программное обеспечение.

Современное развитие электронного обучения (далее – ЭО) в России и в мире в целом претерпевает довольно мощные и интенсивные изменения в рамках смещения акцентов как в русле психолого-педагогического аспекта использования ЭО, так и технического аспекта его реализации. Так, если еще десять лет назад ЭО рассматривалось в образовании только как дополнительное средство организации учебного процесса, то сейчас его уже можно отнести к необходимому компоненту системы обучения [1,2].

В техническом аспекте реализации ЭО ранее наблюдалась тенденция разработки как программного обеспечения, так и самого контента в закрытом, защищенном виде; на текущий момент – видна тенденция к открытости, массовости: например, широкое использование свободного ПО (Moodle, Web 2.0), MOOCs (массовые онлайн курсы на платформах Coursera, EdХи др.).

На наш взгляд, вектор развития в области свободы и открытости всех компонентов ЭО будет в дальнейшем только развиваться и набирать обороты, точно так же, как это происходило с историей свободного программного обеспечения (далее – СПО).

Историю возникновения первого свободного программного обеспечения можно отнести к 60 годам прошлого века, когда IBM выпустила ОС «Airline Control Program» с открытым исходным

кодом. Однако в связи с иском в 67 году со стороны правительства США, в котором указывалось, что компания препятствует развитию честной конкуренции в сфере разработки ПО, идея СПО была отодвинута на несколько десятилетий. И только в 83 году Ричард Столлман объявил о создании фонда свободного программного обеспечения и разработал лицензию GNU. В основе идеи СПО лежит 4 свободы:

- 0 – свобода выполнять программу как угодно для любых целей;
- 1 – свобода изучать и изменять программу;
- 2 – свобода передавать копии программ;
- 3 – свобода передавать копии своих измененных версий программы.

Таким образом, в основе современного понимания сущности СПО лежит наличие открытого программного кода, с возможностью его использования, изучения, адаптации, переработки и дальнейшего распространения программы и не включает требование бесплатности.

Возвращаясь к вопросу об использовании СПО в образовании, следует обратить внимание на распоряжении от 17 декабря 2010 г. № 2299-р «Об утверждении плана перехода федеральных органов исполнительной власти и федеральных бюджетных учреждений на использование свободного программного обеспечения (2011 - 2015 годы)», где указано о необходимости разработки рекомендаций для образовательных учреждений высшего профессионального образования о замене используемого в учебном процессе проприетарного программного обеспечения аналогичным свободным программным обеспечением [3]. Причем, следует отметить, что в сфере образования в идею СПО вкладывают требование бесплатности (которое, как уже было указано выше, отсутствует в определении СПО Р. Столлмана). Таким образом, использование СПО в рамках ЭО является не только технически оправданным, но и законодательно обоснованным выбором для всех образовательных учреждений [6].

В рамках нашего исследования был проведен анализ существующего СПО для поддержки ЭО. В ходе анализа были выделено 8 категорий программ по видам деятельности. В каждой категории программные средства были разделены на те, которые удовлетворяют требованиям СПО, и бесплатные программы, причем первые две категории относятся к базовым решениям СПО, остальные – к дополнительным инструментариям в поддержку ЭО.

1. Управление электронным обучением/учебным контентом (LMS/LCMS):

СПО:

- Sakai - онлайн система организации учебного образовательного пространства. В систему интегрирована поддержка стандартов и спецификаций IMS Common Cartridge, SCORM.

- Moodle - среда дистанционного обучения, предназначенная для создания дистанционных курсов.

- Claroline (Classroom Online) – платформа построения сайтов дистанционного обучения, созданная с учетом пожеланий преподавателей.

- Dokeos – платформа построения сайтов дистанционного обучения, основанная на ветке (fork) Claroline (версии 1.4.2.). Оно подойдет скорее организациям, чем университетам.

- KEEP Toolkit - система электронного обучения и развития человеческих ресурсов с открытым кодом (PHP) eFront обеспечивает возможности Веб 2.0.

- Atutor - система управления обучения с открытым исходным кодом, использует MySQL и PHP, имеет модульную архитектуру.

Бесплатные:

- Moodle – описана выше.

- Claroline – описана выше.

- Dokeos – описана выше [4].

2.Разработка учебных объектов для ЭО

СПО:

- Salasaga - Flash-базирующая среда разработки учебных ресурсов с открытым кодом, написанная на языке C, обеспечивает высокую степень оптимизации автоматически сгенерированного Flash-кода.

- XERTE - среда разработки интерактивных мультимедийных учебных материалов с открытым кодом, содержащая инструменты для разработчиков средств электронного обучения.

- eXe - eLearning XHTML editor - программа для создания курсов электронного обучения. Базируется на веб-браузере Mozilla FireFox. Возможно включение в курсы практически любого материала – веб-страниц, звука, флэш-роликов, видео, рисунков. Поддерживаемая версия – SCORM v. 1.2 .

- Reload Editor - редактор метаданных SCORM-пакета. Позволяет выстраивать структуру курса, создавать и редактировать метаданные курса. Помимо возможностей редактирования курсов, имеет также встроенный эмулятор LMS-системы, что позволяет тут же проверить работоспособность пакета.

Бесплатные:

- QuizForge - это бесплатный набор инструментов для автоматичного генерирования Flash-базирующих учебных тестов и оценочной информации, который не требует написания программного кода.

- XERTE -описана выше.

- eXe -описана выше.
- Reload Editor -описана выше.

3.Реализация инфографики

СПО:

- Inkscape - редактор векторной графики, с возможностями похожими на Illustrator, CorelDraw или Xara X, используя стандарт W3C Scalable Vector Graphics (SVG).

- Stat Planet StatPlanet (ранее StatPlanet Map Maker) - СПО для создания интерактивных карт, которые полностью настраиваемы. Помимо карт, программное обеспечение также имеет возможность создавать интерактивные графики и диаграммы.

Бесплатные:

- amCharts Visual Editor - редактор позволяет использовать amCharts как веб – сервис, функционал: создание различного вида диаграмм, графиков, генерация их в HTML-код и публикации на страницу HTML.

- Dipity - редактор для создания интерактивных, визуально привлекательных временных лент, возможность использования динамических инструментов визуализации с отображением фотографий, видео, новостей и блогов в хронологическом порядке.

- Gliffy- веб-редактор для диаграмм, блок-схем, дизайна пользовательского интерфейса и др.

- Google Chart Tools предоставляет несколько инструментов для построения графиков и диаграмм Chart API.

- Piktochart –позволяет оформить информацию в наглядном, привлекательном виде, имеет большой банк графических элементов.

- Venngage сервис для оформления данных в наглядном виде, наличие сотни профессиональных шаблонов для инфографики, отчетов, плакатов, рекламных акций и т.д.

- Wordle сервис для генерации "облака слов" из текста, возможность настроить облака с различными шрифтами, наличие макетов и цветовых схем.

4.Реализация голосового набора текста

СПО:

- Chrome Speak обеспечивает встроенную поддержку речи Windows (с использованием SAPI 5), Mac OS X и Chrome OS, используя возможности синтеза речи, предоставляемые операционной системой.

Бесплатные:

- Balabolka- программа Text-To-Speech (TTS). Текст может быть сохранен в форматах WAV, MP3, MP4, OGG или WMA. Программа умеет читать вслух содержимое буфера обмена, показывать текст в AZW,

CHM, DjVu, DOC, EPUB, FB2, HTML, LIT, MOBI, ODT, KHP, PDF и RTF файлах.

- FoxVox «говорит» любой текст, который выделен на веб - странице. Может также создавать аудиокниги в форматах MP3, OGG и WAV.

- Free Natural преобразует текст из Microsoft Word, веб - страниц, PDF - файлов и электронной почты в произносимые слова.

- PowerTalk автоматически проговаривает текст из презентаций в Microsoft PowerPoint для Windows. Преимуществом по сравнению с другими программами Text To Speech является то, что PowerTalk умеет говорить скрытый текст, прикрепленный к изображениям.

- QR Code позволяет пользователям преобразовать текст в речь, генерирует QR - код для речевого URL.

- SPEAKIT преобразует текст в речь, читает выделенный текст с использованием технологии Text-To-Speech с автоматическим определением языка, может читать текст на более чем 50 языках.

- TTS Firefox – читает выделенный текст, возможность скачать полученное аудио.

- Voki сервис, который позволяет создавать собственные аватары, добавить голос к аватарам, опубликовать Voki на любой блог, сайт или профиль, а также воспользоваться учебными ресурсами Voki.

- WordTalk - плагин преобразования текста в речь для Microsoft Word, говорит текст документа и подсвечивает его [5].

5.Создание цифрового сторителлинга

Бесплатные:

- Animoto - инструмент для создания видео и презентаций.

- BublblrBublblr представляет собой инструмент для создания комиксов с использованием фотографий с flickr.com.

- Cartoonist - интернет - инструмент для создания мультфильмов или персональные цифровые истории, состоящие из профессиональных фонов, персонажей, реквизита, изображений и текста.

- Comic Master позволит создать свой собственный короткий графический роман.

- MakeBeliefsComix – позволяет создать свой собственный комикс.

- Smilebox позволяет быстро создавать слайд - шоу, приглашения, поздравления, коллажи, и фотоальбомы на компьютере, более 1000 настраиваемых шаблонов на выбор.

6.Организация опросов, голосований, тестирования

СПО:

- OrbeonForms - решение для создания и развертывания веб - форм, обрабатывает сложные формы, реализует стандарт W3C XForms.

Бесплатные

- addpoll - возможность создавать и проводить опросы на компьютере, создавать он-лайн опросы.
- formsite - позволяет создать практически любой тип формы опроса или голосования, имеет 40 видов форм вопросов.
- GFXPOLL.com - ресурс для создания бесплатных графических опросов для сайта, форума или электронной почты.
- Google Формы – создание и проведение опросов, результаты организованы в виде электронной таблицы.
- Scattervox - новый вид опроса, ответы откладываются на двухмерном графике, реализация интерактивной инфографики.
- SISsurveyCreate - бесплатные онлайн-опросы и опросы с использованием SiS обследования.
- SurveyMonkey – сервис для создания опросов, анкетирования.
- EasyTestMaker - онлайн - тестовый генератор
- Hot Potatoes – создание тестов с различными видами тестовых заданий, поддержка стандарта Scorm.

7.Создание скринкастов

СПО:

- RecordMyDesktop — бесплатная утилита командной строки с открытым исходным кодом для записи скринкастов. Результат сохраняется в контейнереogg. Работает в Linux.

Бесплатные:

- FastStone Capture - полнофункциональный инструмент для захвата экрана, который позволяет легко захватывать и делать аннотации на экране, включая окна, объекты, меню, также позволяет записывать действия на экране, звук.
- Camstudio способен записывать экран рабочего стола компьютера, включая аудио-поток, создавать стандартные AVI видео файлы.
- Screencast-O-Matic – это онлайнновый сервис для видеозахвата содержимого экрана, сервис независим от операционной системы, требуется поддержка браузера Java.

8.Организация веб-конференций

СПО:

- OpenMCU-ru - бесплатный программный сервер видеоконференций (программный MCU) с открытым исходным кодом.
- BigBlueButton - система веб-конференций для онлайн-обучения, позволяет обмениваться документами (PDF, любые офисные документы), возможно записывать сессии для последующего воспроизведения.

Бесплатные:

- Google+ Hangouts организация группового чат видео (до 10 человек) бесплатно.
- Conferendo - бесплатные видеозвонки и видеоконференции 3-на-3 HQ качества, возможность проводить бесплатные видеовещания на 20 пользователей.
- TrueConf Server Free - сервер видеосвязи на 6 абонентов; клиентские приложения для всех платформ; защищенная связь внутри сети.

Таким образом, анализ существующего СПО в поддержку ЭО показал наличие широкого спектра возможностей, предоставляемых подобными программами, которые не только расширяют психолого-педагогические, методические возможности реализации ЭО для преподавателей и учащихся, но и возможность дальнейшего развития технических средств реализации ЭО.

Список использованных источников

1. Романов Е.П. Подготовка учителя к использованию Интернет-технологий в профессиональной деятельности: монография. Магнитогорск: МаГУ, 2008. 129с.
2. Боброва И.И. Место и функции дидактических средств информационно-коммуникационных технологий в учебном процессе // Дистанционное обучение: возможности, проблемы и решения: сб. науч. ст. рег. науч.-практ. конф. Магнитогорск: МаГУ, 2011. С. 52-57.
3. Распоряжение Правительства РФ от 17.12.2010 N 2299-р «Об утверждении плана перехода федеральных органов исполнительной власти и федеральных бюджетных учреждений на использование свободного программного обеспечения на 2011 - 2015 годы» [электронный ресурс] – Режим доступа: http://www.consultant.ru/document/cons_doc_LAW_111346/
4. Карманова Е.В. Организация учебного процесса с использованием дистанционных образовательных технологий : [Электронный ресурс] : учебное пособие / Е. В.Карманова; ФГБОУ ВПО «Магнитогорский государственный технический университет им. Г.И. Носова». – Электрон.текстовые дан. (2,23 Мб). – Магнитогорск : ФГБОУ ВПО «МГТУ», 2016.
5. 321 Free Tools for Teachers - Free Educational Technology [Электронныйресурс] - Режим доступа: <http://elearningindustry.com/321-free-tools-for-teachers-free-educational-technology>
6. Чистяков Д.А. К вопросу о реализации электронного обучения средствами Microsoft Sharepoint / Д.А. Чистяков, Л.В. Курзаева // Актуальные проблемы развития вертикальной интеграции системы образования, науки и бизнеса: экономические, правовые и социальные

аспекты: материалы II Международной научно-практической конференции. - 2014. - С. 147-153.

7. Гаврилова И.В., Давлеткиреева Л.З. Всероссийская молодёжная конференция с элементами научной школы «Теория и практика применения свободного программного обеспечения» // Теория и практика применения свободного программного обеспечения. – Магнитогорск: МаГУ, 2011. – С. 4-16

Построение онтологической модели организации образовательного процесса в системе Protege

*Козлова Е.С.,
ФГБОУ ВО «Магнитогорский государственный университет
им. Г.И. Носова»*

Аннотация. В статье приводится описание процесса реализации онтологии, описывающей организацию образовательного процесса в среде Protégé. Выделены основные классы и связи между ними, представлены примеры запросов к онтологии.

Ключевые слова: онтология, образование, Protégé.

Развитие и популяризация дистанционных технологий обуславливает поиск путей эффективной организации электронного обучения.

Одним из перспективных направлений является онтологическое моделирование предметной области. По своей сути онтология предметной области представляет собой формальную модель понятийной структуры предметной области. С методической точки зрения этот подход – один из наиболее «систематических» и наглядных. С технологической – открытый подход для внесения изменений с минимальным риском получения противоречивых данных и знаний, что особенно актуально для образовательных задач.

Рассмотрим создание такой модели для системы обучения с использованием дистанционных технологий в системе Protege.

Protege – это Java-программа, которая предназначена для создания онтологий различных предметных областей. Этапами построения онтологической модели выступают:

- 1) создание классов и подклассов;
- 2) создание слотов;
- 3) создание экземпляров классов;
- 4) настройка форм отображения;
- 5) формирование экземпляров классов;

6) формирование запросов.

Начнем с первого этапа: создание классов и подклассов. Классы являются основными категориями предметной области, над которыми производятся действия. Для электронного обучения мы выделили следующие классы: стандарт; учебный план; дисциплина; разделы дисциплины; дисциплина-оценка; студент; оценка возможная.

Для создания классов/подклассов в Protégé мы воспользовались вкладкой Classes и операцией создание класса, а, чтобы создать подкласс необходимо проделать, все то же самое только при этом выбрать класс-родитель. На рисунке 1 продемонстрированы основные операции по созданию класса/подкласса.

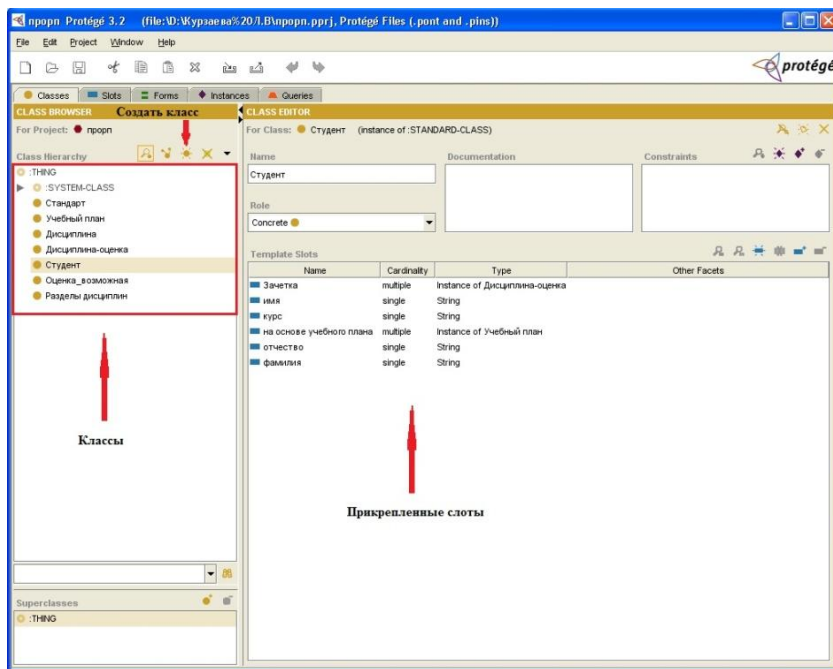


Рис. 1. Классы онтологической модели

На втором этапе осуществляется работа со слотами. В онтологических моделях классами являются общие понятия предметной области, которые обычно представлены иерархично. Каждый класс описывает индивидуальность сущностей, которые объединены по общим атрибутам или свойствам, например, уровень знаний студента или курс его обучения. Атрибуты и отношения класса описываются конструкцией под названием слот (атрибуты, свойства).

В рамках данного этапа нами были созданы основные свойства классов. На рисунке 2 представлены слоты классов в программном средстве Protege на примере слота «Входят в». Для того чтобы создать слот нужно перейти на вкладку Slot и воспользоваться операцией создание слота. Также каждый атрибут имеет ограничения, такие как: тип значения (численный, строчный, логический) и др. пределы, которые выражаются в максимальном и минимальном значении.

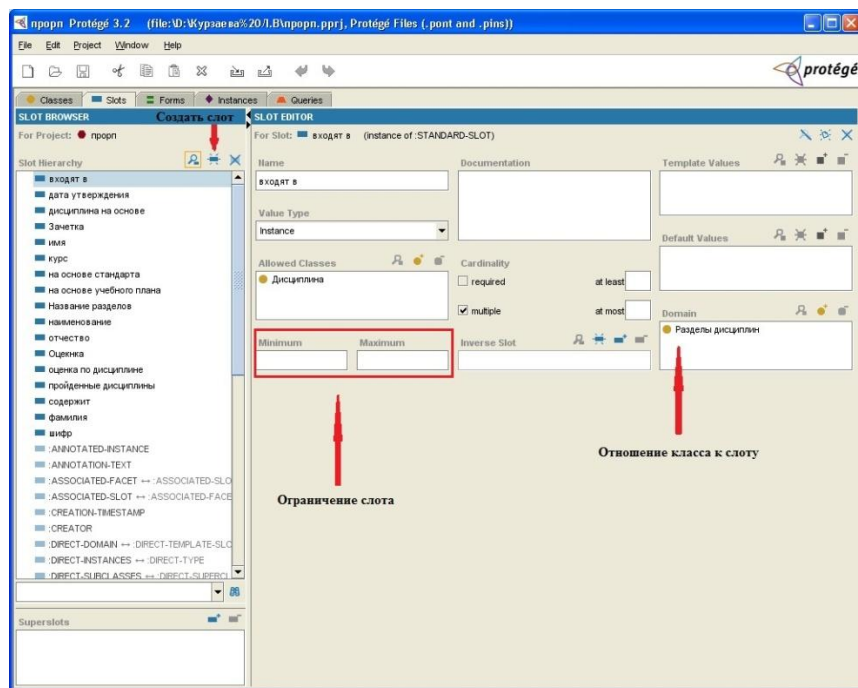


Рис. 2. Свойства классов

Каждый из ограничений будет задействован в системе Protege при создании слотов. Мы воспользовались такими типами как:

- Instance – тип отношений;
- String – текстовое значение;

Также можно установить минимальный и максимальный пределы. В системе Protege для описания ограничений нужно воспользоваться панелью SlotEditor (рисунок 2).

Следующий этап — это создание отношений между классами. Отношения создаются как слоты, именуются глаголом и указывают на

связь между классами. На рисунке 3 представлено отношение на примере слота «Оценка по дисциплине».

Этап создания экземпляров классов заключается в формировании базы знаний. Перед тем, как начать формировать базу знаний необходимо перепроверить структуру созданной онтологической модели, иначе изменения структур может повлечь потерю введенных данных. Кроме того, при добавлении новых слотов, необходимо заполнять их значения для старых экземпляров классов [16]. Далее производится настройка форм отображения. Для каждого класса в системе формируется форма отображения, которая будет использоваться для ввода данных экземпляра. Поля соответствуют слотам [16]. Для разных типов данных слотов существуют разные типы полей для ввода данных [16]: текстовый (TextFieldWidget), целочисленный (IntegerFieldWidget/FloatFieldWidget), список экземпляров (InstanceListWidget/InstanceTableWidget).

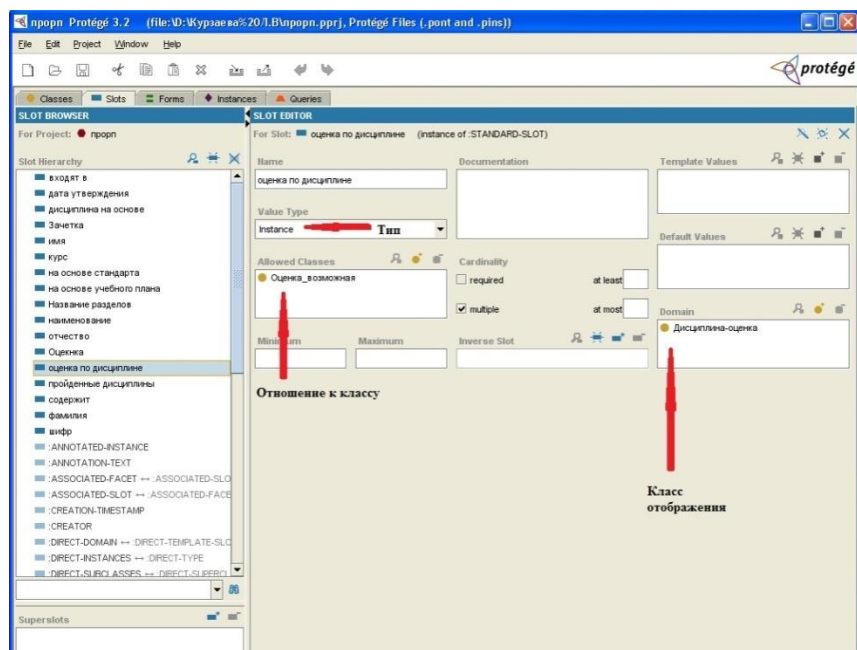


Рис. 3. Отношение между классами

На рисунках 4, 5 представлены слоты на вкладке «Forms», на примере класса «Дисциплина».

На этапе формирования экземпляров класса мы создадим экземпляры для классов «Дисциплина» и «Дисциплина-оценка» продемонстрированные на рисунках 6, 7.

Для того, чтобы создать экземпляр класса «Дисциплина-оценка» необходимо внести оценку дисциплины, и также привязать ее к классу «Дисциплина», так как каждая оценка соответствует определенной дисциплине (рисунок 5).

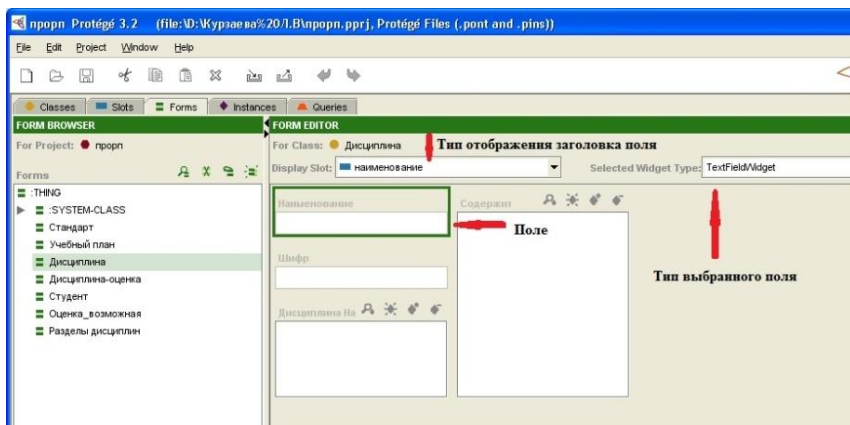


Рис. 4. Настроенная форма класса «Дисциплина» слота «Наименование»

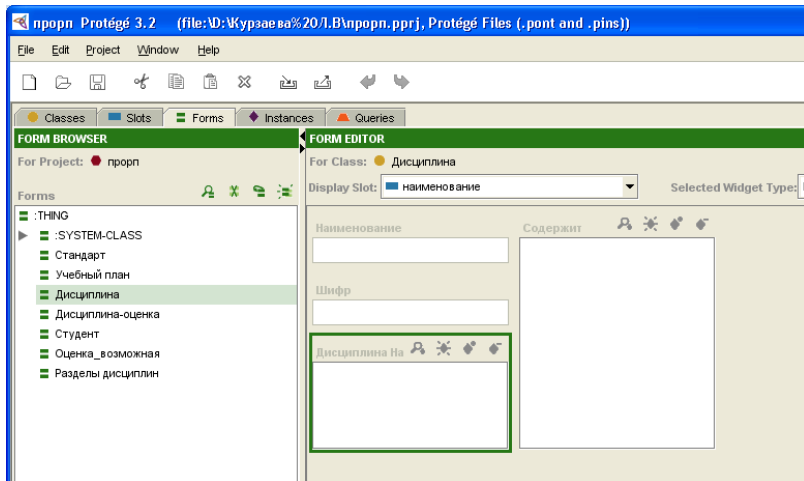


Рис. 5. Настроенная форма класса «Дисциплина» слота «Дисциплина на основе»

Запросом формируется условие поиска. На данном этапе создадим несколько запросов:

- На основе какого учебного плана прописан стандарт образовательной программы «Бизнес-информатика».
- Какие разделы входят в дисциплину «Управление проектами».
- У каких студентов в зачетке стоит «отлично» по дисциплине «Управление проектами».

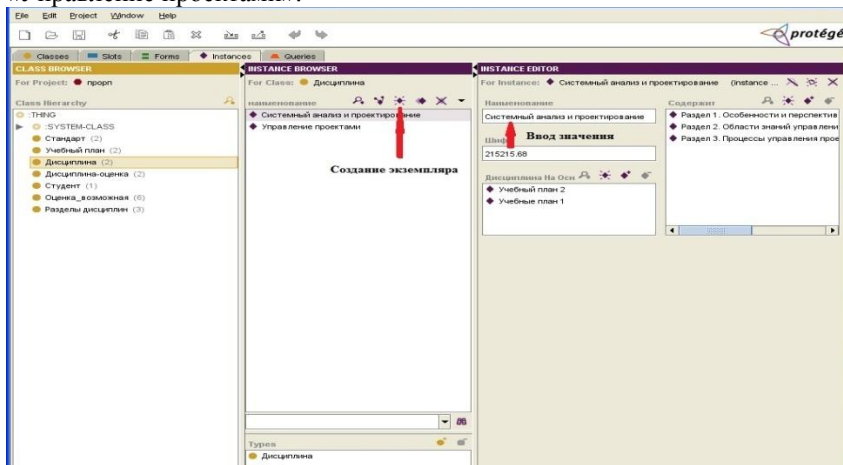


Рис. 6. Экземпляры класса «Дисциплина»

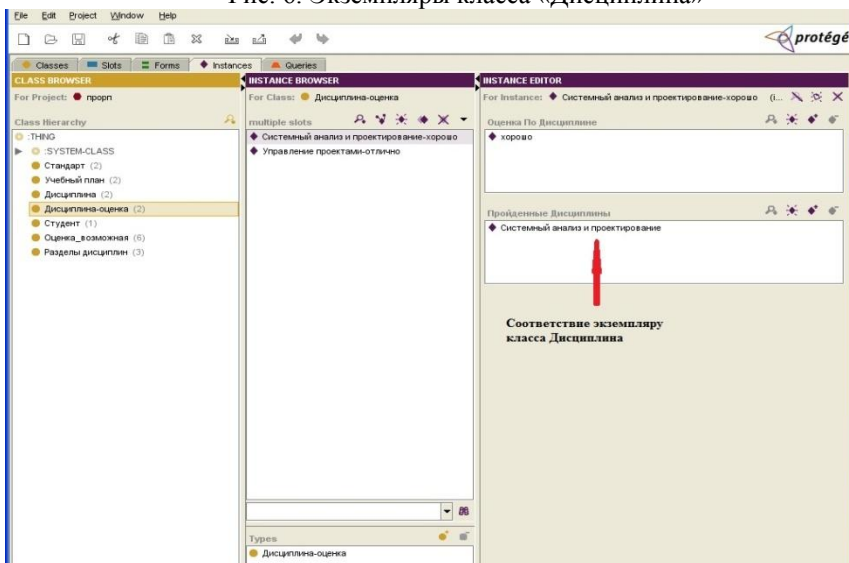


Рис. 7. Экземпляры класса «Дисциплина-оценка»

Для создания запросов необходимо перейти на вкладку «Queries» и заполнить следующие поля (рисунок 8):

- Class. Определение класса для запроса.
- Slot. Определение слота в соответствии с выбранным классом.

Условие:

- contains (содержит);
- does not contains (не содержит);
- is (является или равно);
- is greater then (более чем, где значение условия не включается);
- is less than (менее чем, где значение условия не включается).

Float. Значение, удовлетворяющее условию и запросу.

Созданные запросы представлены на рисунках 8,9,10.

В системе можно формировать неограниченное количество запросов с различными вариациями условий. Созданные условия можно сохранять для дальнейшего использования.

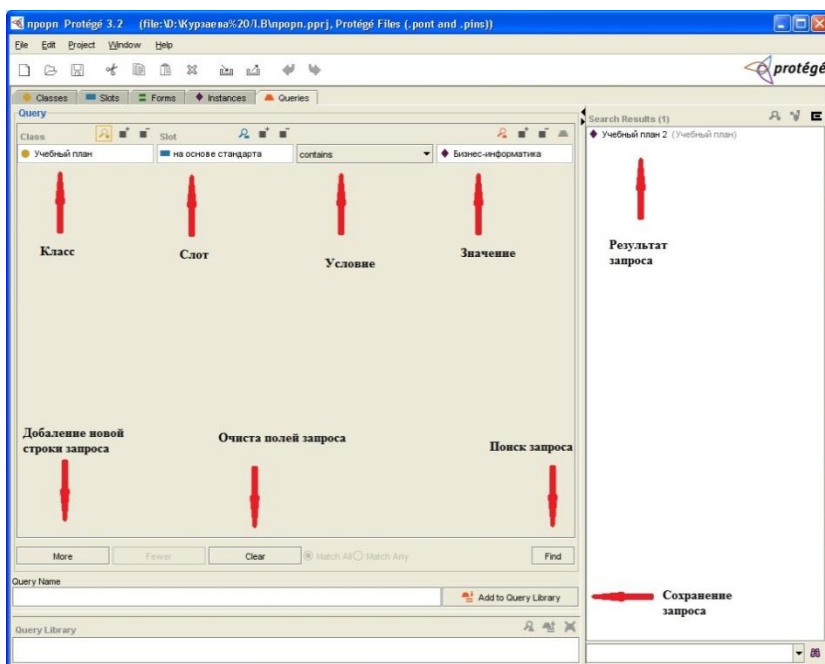


Рис. 8. Создание запроса

Полученная онтологическая модель, по своей сути, является прототипом базы знаний по организации обучения с использованием дистанционных технологий. Следует отметить, что процесс создания онтологий итерационный, а построенная модель является только началом

в процессе описания рассматриваемой предметной области и нуждается в доработке.

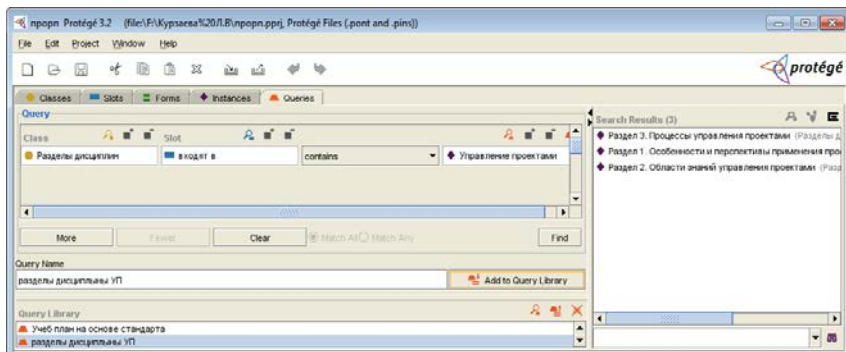


Рис. 9. Создание запроса: «Какие разделы входят в дисциплину «Управление проектами»

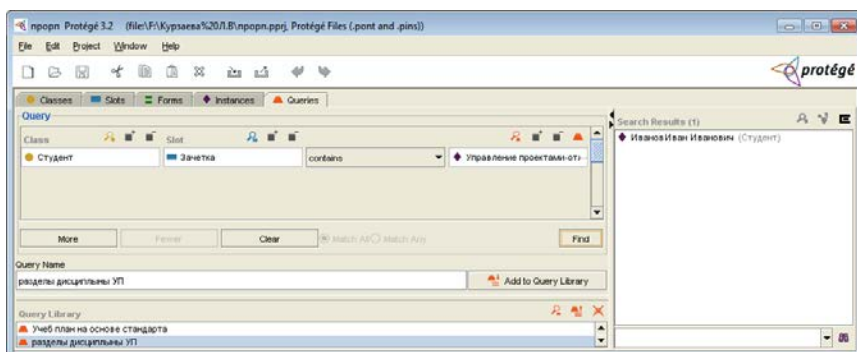


Рис. 10. Создание запроса: «У каких студентов в зачетке стоит «отлично» по дисциплине «Управлению проектами»

Список использованных источников

1. Белоусова И.Д. К вопросу о согласовании требований к содержанию профессиональной подготовки на основе онтологической модели / И.Д. Белоусова, Л.В. Курзаева, А.М. Агдавлетова // Современные наукоемкие технологии. - 2015. - № 11. - С. 67-70.

2. Белоусова И.Д., Курзаева Л.В., Лактионова Ю.С., Агдавлетова А.М. Онтологическая модель управления требованиями в процессе подготовки ИТ-специалистов // Успехи современной науки. -2016. -Т. 1.- № 3. -С. 98-100.

3. Курзаева, Л.В. Введение в теорию систем и системный анализ: учеб. Пособие / Л.В. Курзаева. -Магнитогорск: МГТУ им. Г.И.Носова, 2015. -211 с.

4. Курзаева Л.В. Структурно-функциональная модель развития конкурентоспособности будущего ИТ-специалиста в процессе профессиональной подготовки в вузе: организационно-управленческий аспект/Л.В. Курзаева//Современные проблемы науки и образования. - 2012. - № 6; Режим доступа: <http://www.science-education.ru/106-7424> (дата обращения 12.06.2015).

5. Курзаева Л.В. Дистанционный курс «Инструментальные методы поддержки принятия решений»: электронный учебно - методический комплекс//Хроники объединенного фонда электронных ресурсов Наука и образование. -2016. -№ 1 (80).- С. 2.

6. Курзаева Л.В. Совершенствование методов идентификации требований к результатам профессионального обучения на основе экспертной информации /Л.В. Курзаева// Информатизация образования и науки. - 2015. - № 4 (28). - С. 167-175.

7. Козлова Е.С. Обоснование выбора системы дистанционного обучения в высшем учебном заведении /Козлова Е.С., Черкасов М.А., Макашова В.Н., Курзаева Л.В.// В сборнике: Современные проблемы развития фундаментальных и прикладных наук Материалы II международной научно-практической конференции. -2016. -С. 48-56.

8. Черкасов М.А. К вопросу о принятии решений по управлению качеством дистанционного образования на основе метода анализа иерархий / Черкасов М.А., Козлова Е.С., Курзаева Л.В., Макашова В.Н.// Успехи современной науки и образования. -2016. -Т. 1. -№ 3. -С. 121-124.

Онтологическое моделирование содержания профессиональных компетенций в среде «Protege»

*Курзаева Л. В., к.п.н.,
Соколова А. А., Кириллов Д. В.,
ФГБОУ ВО «Магнитогорский государственный технический
университет им. Г.И. Носова*

Аннотация. Рассмотрены понятия «онтология» и «профессиональные компетенции». Приведен пример построения онтологической модели содержания профессиональных компетенций для кафедры прикладной информатики в среде Protege, представлено подробное описание данного программного средства.

Ключевые слова: онтология, компетенция, Protege, моделирование.

Современная компетентностная парадигма в образовании требует тщательной проработки требований к содержанию и уровню результатов обучения выпускников учебных заведений. Существует огромное количество трактовок данных требований к той или иной компетенции со стороны образовательных и профессиональных стандартов, рамок компетенций. В связи с этим, все сложнее становится процесс их гармонизации и учета, а также определения дисциплин, которые их формируют.

В век развитой информатизации и компьютеризации существуют широкие возможности для осуществления процесса формирования профессиональных компетенций. Данные возможности выражаются в большом количестве методов, а самое главное средств, которые помогают моделировать, анализировать и упрощать проведения анализа в той или иной области.

В данной статье будет рассмотрен один из таких методов – метод онтологического моделирования, проведенный с помощью автоматизированного средства Protege 4.

Цель данной статьи заключается в исследовании процесса онтологического моделирования содержания профессиональных компетенций с помощью свободно распространяемого средства Protege 4.

Метод онтологического моделирования достаточно сложен в использовании. Онтологии – больше чем просто сложный подход к описанию и классификации информации. Для рассмотрения метода онтологического моделирования содержания профессиональных компетенций необходимо определиться с тем, что же такое «Онтологии».

Онтологии используются для описания знаний о некоторой предметной области. Онтологии описывают понятия предметной области, а также отношения, которые существуют между этими понятиями. Различные онтологические языки предоставляют различные возможности. Самые последние разработки стандартных языков описания онтологии – OWL из World Wide Web Consortium (W3C) [1].

Онтологии позволяют представить новые понятия так, что они становятся пригодными для машинной обработки. С помощью онтологии можно "перекинуть мостик" между новыми понятиями, с которыми система еще не встречалась, и описаниями уже известных классов, отношений, свойств и объектов реального мира [2].

Компоненты, из которых состоят онтологии, зависят от парадигмы представления. Но практически все модели онтологий в той или иной степени содержат концепты (понятия, классы, сущности, категории), свойства концептов (слоты, атрибуты, роли), отношения между

концептами (связи, зависимости, функции) и дополнительные ограничения (определяются аксиомами, в некоторых парадигмах фасетами) [2].

Protege – локальная, свободно распространяемая Java-программа, разработанная группой медицинской информатики Стэнфордского университета. Программа предназначена для построения (создания, редактирования и просмотра) онтологий прикладной области. Её первоначальная цель – помочь разработчикам программного обеспечения в создании и поддержке явных моделей предметной области и включение этих моделей непосредственно в программный код. Protege включает редактор онтологий, позволяющий проектировать онтологии разворачивая иерархическую структуру абстрактных или конкретных классов и слотов. Структура онтологии сделана аналогично иерархической структуре каталога. На основе сформированной онтологии, Protege может генерировать формы получения знаний для введения экземпляров классов и подклассов. Инструмент имеет графический интерфейс, удобный для использования неопытными пользователями, снабжен справками и примерами [3].

Protege основан на фреймовой модели представления знания ОКВС (Open Knowledge Base Connectivity) [4] и снабжен рядом плагинов [5], что позволяет его адаптировать для редактирования моделей, хранимых в разных форматах (стандартный текстовый, в базе данных JDBC, UML, языков XML, XOL, DAML+OIL, OWL и другие.

Protege OWL позволяет описывать не только понятия, но и конкретные объекты. Он имеет богатый набор операторов – например, пересечение, объединение и отрицание. Он основан на логической модели, которая позволяет создавать определения, соответствующие неформальному описанию [1].

Таким образом, определения сложных понятий могут быть построены на основе определений более простых понятий. Кроме того, логическая модель позволяет использовать рассуждения, которые могут проверить все ли утверждения и определения в онтологии взаимно согласуются и могут также выяснить, какие концепции соответствуют заданными определениям [1].

Для того, чтобы тщательно построить онтологическую модель содержания профессиональных компетенций кафедры прикладной информатики, необходимо также разобраться с тем, в чем же заключается суть профессиональных компетенций.

Сегодня термином профессиональные компетенции обозначаются способности успешно действовать на основе практического опыта, умений и знаний при решении задач профессионального рода деятельности. Профессиональные компетенции следует отличать от общих (универсальных) компетенций, которые определяются как базовое

качество индивида, которое определяющим образом влияет на эффективное выполнение им рабочих заданий в определенных ситуациях. Поскольку управление профессиональными компетенциями немисливо без системы стандартов, то можно дать следующее определение профессиональной компетенции [6,7].

Профессиональные компетенции – это способности работника выполнять работу в соответствии с требованиями должности, а требования должности – задачи и стандарты их выполнения, принятые в организации или отрасли [6, 8].

Для определения профессиональной компетенции для кафедры прикладной информатики выделены следующие параметры:

1. Компетенция.
2. Дисциплина.
3. Содержание.
4. Уровни.

Содержание компетенции делится еще на три составляющих: знать, уметь и владеть. Другими словами, чтобы определить ту или иную компетенцию выпускника кафедры прикладной информатики для конкретной дисциплины, необходимо знать определенные термины, уметь выполнять конкретные действия и владеть определенными навыками.

Уровни также делятся на три вида: пороговый, средний и высокий. Это означает, что у каждой компетенции для конкретной дисциплины есть свой соответствующий уровень, который характеризуется соответствующим содержанием [9,10, 11].

По данным параметрам можно определить, из чего формируется та или иная компетенция. Если связать все вышеупомянутые параметры, то по конкретным запросам можно сформировать или определить необходимую компетенцию, которая в свою очередь относится к конкретной дисциплине. Для более удобного формирования или определения таких компетенций и было использовано онтологическое моделирование в среде Protege 4.

Весь процесс онтологического моделирования содержания профессиональных компетенций в среде Protege 4 был выполнен по специальному справочнику «Практический справочник по построению Онтологий OWL в Protege 4» [1]. Данный справочник задает определенную последовательность действий, а именно (кратко):

1. Открыть программное средство Protege 4.
2. Создать классы предметной области.
3. Создать подклассы для классов предметной области.
4. Создать связи между классами и подклассами.
5. Сформировать модель онтологий.
6. Прописать индивиды классов и подклассов.

7. Прописать запросы для получения результата.

Далее рассмотрим подробнее каждый из этапов онтологического моделирования содержания профессиональных компетенций в среде Protege 4.

Перед тем, как создавать онтологическую модель необходимо тщательно проанализировать предметную область. Будучи полностью готовым к работе с программным средством, выполняем очевидный шаг – открываем программу Protege 4.2.

После того, как мы открыли программу, переходим к самой главной части онтологического моделирования – определяем и создаем классы. Классы – это основные элементы онтологии OWL. В Protege 4 создание классов осуществляется в закладке «Classes». Вышеупомянутые параметры для формирования профессиональных компетенций будут являться классами для нашей онтологической модели. Составляющие параметров будут являться подклассами классов. Имена классов и подклассов рекомендовано задавать с помощью латинских букв, это устранил ряд возникающих ошибок в работе приложения. Класс THING - это класс, представляющий набор, содержащий все объекты предметной области. На рисунке 1 представлена схема классов для нашей предметной области.

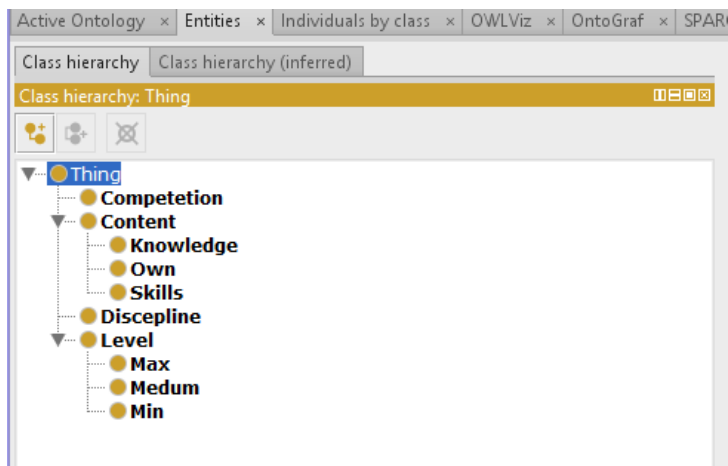


Рис. 4. Схема классов онтологической модели содержания компетенций в среде Protege

Для каждого класса и подкласса существуют свойства, на основе которых все классы или подклассы связаны между собой. Существует три основных типа свойств: свойства объектов, свойства типа данных (Data Type) и свойства аннотаций. Свойствами объекта являются

отношения между двумя индивидами. Свойства объекта связывают индивидов. Свойства аннотации могут использоваться для добавления информации (метаданные) для классов, отдельных индивидов и свойств объектов/ типов данных. Также свойства бывают прямыми, обратными, функциональными, обратными функциональными и транзитивными.

Для содержания профессиональных компетенций мы будем использовать прямые и обратные свойства. В OWL свойства могут иметь подсвойства, таким образом, чтобы это отражало форму иерархии свойств. Также каждое свойство объекта может иметь соответствующее обратное свойство. Если некоторое свойство связывает индивида *a* с некоторым индивидом *b*, то его обратное свойство связывает индивид *b* с индивидом *a*. Пример свойств для нашей предметной области представлен на рисунке 2. На рисунке представлены свойства класса *Competition* с другими классами и подклассами.

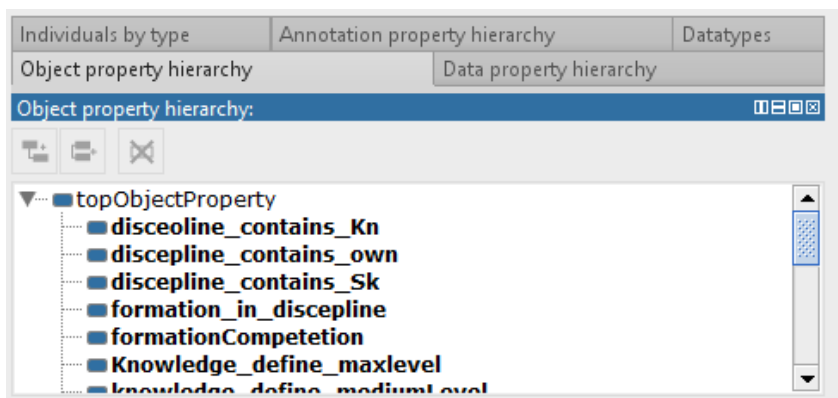


Рис. 5. Свойства класса *Competition* с другими классами

После того, как заданы все свойства для классов и подклассов, можно сформировать онтологическую модель в виде графа. Граф онтологической модели содержания профессиональных компетенций, который показывает свойства классов, представлен на рисунке 3.

Одним из последних, но не менее важных этапов онтологического моделирования для нашей предметной области является создание индивидов классов и подклассов. Данный этап можно охарактеризовать как наполнение контента для классов, прописывание конкретных характеристик, которые и будут в дальнейшем отвечать на поставленные вопросы. На рисунке 4 представлен пример индивидов класса *Content*, подкласса *Skills*.

Также существует возможность прописать для индивидов ограничения по свойству *hasValue*. Ограничение по свойству *hasValue*,

обозначается символом э. Оно описывает набор отдельных лиц, которые имеют, по крайней мере, одно отношение по указанному свойству для конкретного лица.

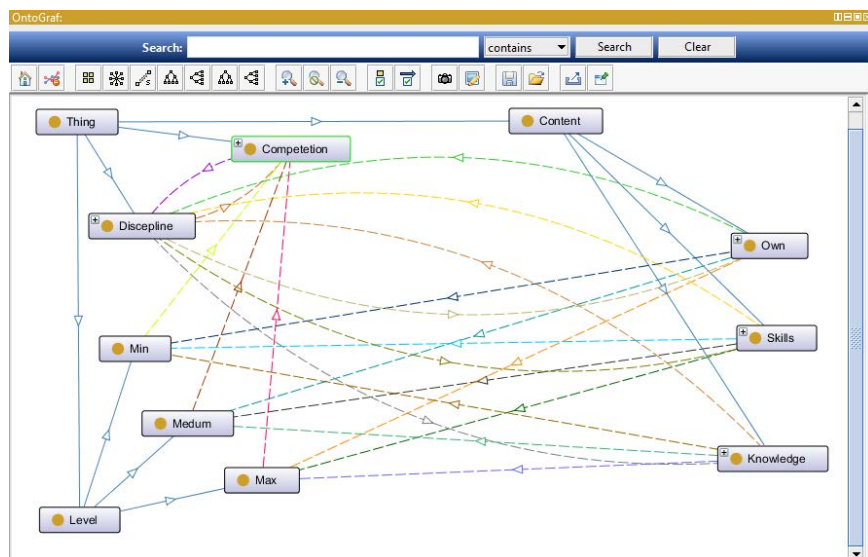


Рис. 6. Онтологический граф содержания профессиональных компетенций

Помимо проделанных нами этапов онтологического моделирования содержания профессиональных компетенций существует большое количество других не менее полезных возможностей для построения онтологий в программном средстве Protege.

Заключительным этапом нашей работы является прописывание запросов, которые и являются результатом для онтологического моделирования. Другими словами, все предыдущие этапы были выполнены для корректного получения результата с помощью запросов. Пример простого запроса для нашей задачи представлен на рисунке 5.

Для составления более сложных запросов необходимо делегирование базы онтологий на сервер, так как в запросе необходимо прописать ссылку на базу. В дальнейшем, с помощью сложных запросов можно будет определять какое содержание требуется для формирования компетенции по конкретной дисциплине на том или ином уровне.

Подводя итоги, можно сказать, что, безусловно, на сегодняшний день любая деятельность значительно упрощается автоматизацией. Так для процесса формирования и анализа профессиональных компетенций выпускников кафедры прикладной информатики был подобран метод

онтологического моделирования с использованием программного средства Protege.

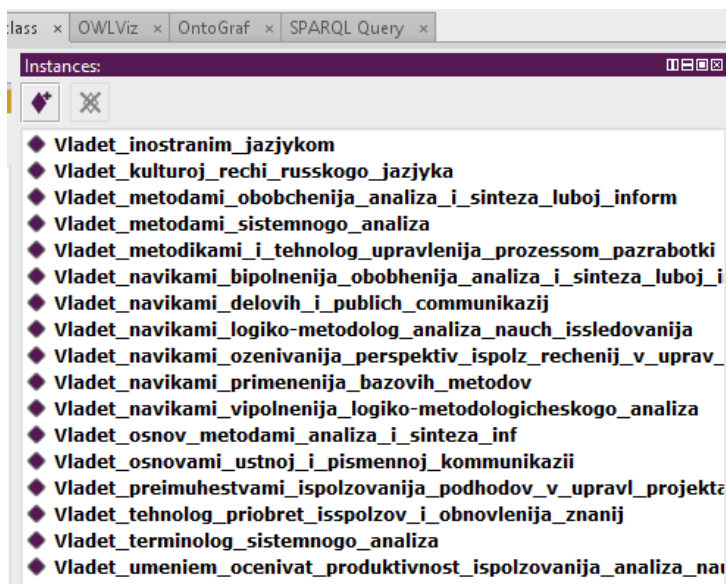


Рис. 7. Индивиды класса Content, подкласса Skills

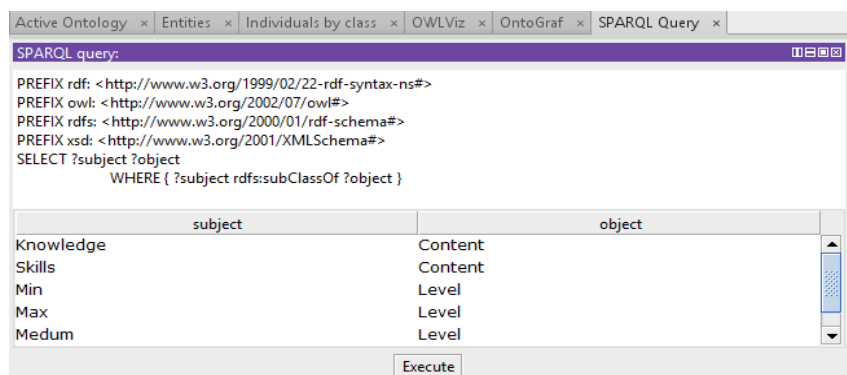


Рис. 8. Пример запроса на вывод подклассов для каждого класса

Результатами создания онтологий будут являться данные, сформированные путем запросов, которые помогут вывести необходимую информацию по заданным свойствам, а именно

сформировать содержание компетенций, которыми должны обладать выпускники, а также подобрать необходимые для этого дисциплины. Реализованный в рассматриваемой версии Protégé подход на основе семантических сетей является не самым простым в использовании. При этом он позволяет не только наглядно представить сложную предметную область, но и создать базу знаний для последующей разработки экспертной системы.

Список использованных источников

1. Ректор А., Стивенс Р., Крис Р. Практический справочник по построению Онтологий OWL в Protege 4. - 2012.
2. Овдей О.М. Обзор инструментов инженерии онтологий / О.М. Овдей, Г.Ю. Проскудина. – Режим доступа: <http://www.interface.ru/home.asp?artId=33246>
3. Мусен. М. Доменные онтологии в области разработки программного обеспечения: Использование Protege с ЭОН Архитектурой // Методы информатики в медицине. - 1998. - с. 540-550.
4. Чаудри В., Фарквар А., Карп Р. П., Райс Дж. Программный фонд базы знаний совместимости // Пятнадцатая Национальный конф. по искусственному интеллекту 1998. - с. 600-607.
5. Ной Н., Синтек М., Декер С., Крабези М., Фергюсон Р., Мусен М. Создание веб-содержимого Semantic с Protege – 2000. // IEEE Интеллектуальные системы. - 2001. - С. 60-71.
6. Белоусова И.Д. К вопросу о согласовании требований к содержанию профессиональной подготовки на основе онтологической модели / И.Д. Белоусова, Л.В. Курзаева, А.М. Агдавлетова // Современные наукоемкие технологии. - 2015. - № 11. - С. 67-70.
7. Белоусова И.Д., Курзаева Л.В., Лактионова Ю.С., Агдавлетова А.М. Онтологическая модель управления требованиями в процессе подготовки ИТ-специалистов // Успехи современной науки. - 2016. -Т. 1. -№ 3. - С. 98-100.
8. Курзаева Л.В. Введение в теорию систем и системный анализ: учеб. Пособие / Л.В. Курзаева. -Магнитогорск: МГТУ им. Г.И.Носова, 2015. -211 с.
9. Курзаева Л.В. Структурно-функциональная модель развития конкурентоспособности будущего ИТ-специалиста в процессе профессиональной подготовки в вузе: организационно-управленческий аспект/Л.В. Курзаева//Современные проблемы науки и образования. - 2012. - № 6; Режим доступа: <http://www.science-education.ru/106-7424> (дата обращения 12.06.2015).
10. Курзаева Л.В. Дистанционный курс «Инструментальные методы поддержки принятия решений»: электронный учебно -

методический комплекс//Хроники объединенного фонда электронных ресурсов Наука и образование. 2016. № 1 (80). С. 2.

11. Курзаева Л.В. Совершенствование методов идентификации требований к результатам профессионального обучения на основе экспертной информации /Л.В. Курзаева// Информатизация образования и науки. - 2015. - № 4 (28). - С. 167-175.

Планирование внеурочного занятия по теме «Информационная безопасность» с использованием среды Scratch

*Предеина О. С., Аркатова К.А.,
ФГБОУ ВО «Магнитогорский государственный технический
университет им Г.И. Носова»*

Аннотация. В статье описана методика проведения внеурочного занятия по теме «Информационная безопасность». Приводится описание интерактивной игры для закрепления учебного материала.

Ключевые слова: Scratch, методика, неурочное занятие, информационная безопасность

Внеурочные занятия представляют собой один из наиболее сложных видов учебного взаимодействия. Проблема как сделать такое занятие интересным и познавательным одновременно является несомненно актуальной. При этом данные занятия должны быть направлены на гармоничное формирование и закрепление предметных, метапредметных и личностных результатов обучения [4, 8,7, 15, 16].

Для проведения таких занятий очень редко используются всевозможные среды программирования. Это объясняется сложностью необходимых предметных знаний, умений и навыков. В этой связи интерес представляет собой среда Scratch [1, 2, 3].

Scratch – это бесплатная среда программирования, ориентированная на учеников 3-5 классов, которая позволяет детям самостоятельно создавать игры, мультфильмы, различные проекты, привлекая к работе все свои творческие навыки и фантазию. Scratch разработан специально для более интересного и увлекательного обучения детей программированию [1, 2, 5].

Рассмотрим планирование внеурочного занятия по теме «Информационная безопасность», целью которого является формирование представления об информационной безопасности с использованием среды Scratch [9-14].

Задачи внеурочного занятия:

- познакомить с понятием информационной безопасности;

- рассмотреть различные угрозы информационной безопасности;
- совершенствовать коммуникативные навыки через умение излагать мысли, умение вести диалог;
- определить план действий для предотвращения угрозы информационной безопасности;
- воспитывать ответственность за свои действия [15].

Учитель: Проблемы защиты информации от постороннего доступа и нежелательного воздействия на нее возникло с той поры, когда человеку по каким-либо причинам не хотелось делиться ею. С переходом на использование технических средств связи, информация подвергается воздействию случайных процессов (неисправностям и сбоям оборудования, ошибкам операторов и т.д.), которые могут привести к ее разрушению, изменению на «ложную», а также создать предпосылки доступа к ней посторонних лиц. Защита информационных процессов выдвигается на первый план.

Вопрос учителя: **Что вы понимаете под термином «информационная безопасность»?**

Дети выдвигают свои варианты ответов.

Обобщая, учитель сообщает определение, которое записывается в тетрадь.

Под **информационной безопасностью** будем понимать защищенность информации от случайных или преднамеренных воздействий характера, которые могут нанести неприемлемый ущерб субъектам информационных отношений, в том числе владельцам и пользователям информации. Хакер – это компьютерный взломщик, который проникает в информационные сети, банки данных и т.д. с целью получить доступ к секретной информации.

Ценной становится та информация, обладание которой позволит ее владельцам получить какой-либо выигрыш.

Вопрос учителя: **Какие воздействия могут нанести ущерб информации, то есть что представляет угрозу информационной безопасности?**

Дети выдвигают свои варианты ответов.

Учитель рассказывает существуют виды воздействия на информационную безопасность, основные моменты записываются в тетрадь.

1. Кража личных данных, утечка информации
2. Вирусы, черви, трояны
3. Спам
4. Хакеры
5. Авторское право, нелицензионное ПО
6. Мошенничество
7. Дезинформация

Учитель: Поговорим теперь о современных методах защиты информации. Учитель сообщает существующие методы защиты и раскрывает их суть, основные моменты записываются в тетрадь.

Для защиты данных применяют различные современные методы, которые могут предотвратить утечку информации, либо ее потерю. Сегодня используется шесть основных способов защиты:

- Препятствие
- Маскировка
- Регламентация
- Управление
- Принуждение
- Побуждение.

Группа методов **Маскировка** — метод защиты информации шифрованием; посторонние лица не смогут ее понять, для расшифровки потребуется знание принципа шифрования.

Управление — способы защиты информации, при которых осуществляется управление над всеми компонентами информационной системы.

Регламентация — важнейший метод защиты информационных систем, предполагает введение особых инструкций, согласно которым должны осуществляться все манипуляции с охраняемыми данными.

Принуждение — методы защиты информации, предполагающие введение комплекса мер, при которых работники вынуждены выполнять установленные правила.

С помощью программы для программирования Scratch мы создадим игру, в которой рассмотрим, как можно защитить свои файлы от посторонних глаз и уберечь их от изменений. На примере такого способа защиты информации, как *регламентация* [6, 7, 8].

Мы создадим игру, сюжет которой заключается в том, чтобы защитить наш компьютер, который содержит секретную информацию, от нежелательного воздействия на него. Воздействовать на компьютер будут различные вирусы. Вирусы будут пробираться к компьютеру, который в свою очередь будет уменьшаться в размере от каждого нежелательного прикосновения. Щитом для защиты нашей информации будет персонаж «Информационная Безопасность», который будет ловить вирусы, тем самым защищая компьютер от вирусов.

Ход работы

1. Создаем новый проект: создаем спрайт «Мой компьютер», в котором будет зашифрована информация.
2. Создадим вирусы, их может быть несколько. Вирусы будут двигаться по полю. Если вирус пробирается к компьютеру, то он наносит ему вред.

3. Нарисуем фон.

4. Создаем самый важный в нашей игре спрайт с именем «Информационная безопасность» в виде щита, безопасность будет защищать наш компьютер от нежелательного воздействия. Сделаем Безопасности программу. При соприкосновении компьютера с вирусом компьютер уменьшается в размере и теряет часть важной информации. Компьютер восстановит свой размер только через 6 секунд. Если компьютер не успевает восстановить свой размер, то он уменьшится еще больше. Наша задача - не дать вирусам пробраться к компьютеру, используя щит «Информационная безопасность».

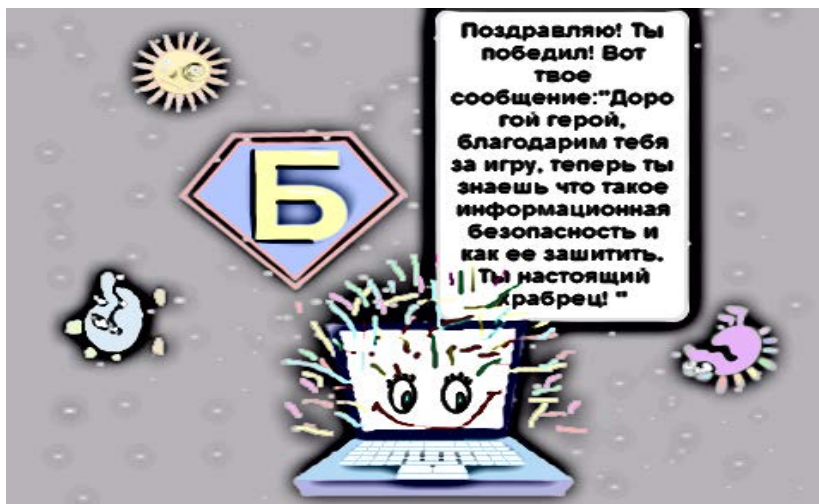


Рис. 1. Вариант фона и спрайтов создаваемой игры

5. Игра длится 5 минут, если компьютер сильно уменьшился, то вирусы нанесли ему вред, и компьютер не может передать нам сообщение. Если компьютер остался в прежнем виде, то вы защитили свой компьютер от вирусов и прошли игру.

6. При выигрыше компьютер улыбается и выводит на экран текст благодарности. При проигрыше компьютер плачет, и можно начать игру заново.

Считаем предложенный способ организации внеурочного занятия весьма перспективным и привлекательным.

Список используемой литературы

1. Scratch. Общедоступное программирование в Scratch. - Режим доступа: <http://scratch.uvk6.info/>

2. Зеркина Е.В., Литвин А.В. Преподавание анимационной среды программирования СКРЕТЧ в системе дополнительного образования (на примере кружка) // Актуальные проблемы прикладной информатики и методики обучения информатике : материалы молодеж. всеросс. науч.-практ. конференции / отв. ред. И.Н. Слинкина. – Шадринск : ШГПИ, 2009. – 265 с. – с. 66-68.

3. Доколин А.С., Чернова Е.В. Влияние практического изучения специализированного программного обеспечения при изучении курса «Информационная безопасность» на формирование профессиональных компетенций / Сборник научных трудов по материалам международной заочной научно-практической конференции «Актуальные направления научных исследований XXI века: теория и практика». – Воронеж : УОП ФГБОУ ВПО «ВГЛТА», 2013. – № 2. – 344 с. - с. 134-139.

4. Курзаева Л.В., Чусавитина Г.Н. Подготовка будущих педагогических кадров к превенции киберэкстремизма среди молодежи: моделирование процесса установления требований к процессу профессиональной подготовки // Фундаментальные исследования. -2014. -№ 12-5. -С. 1078-1082.

5. Курзаева Л.В. Развитие информационно-аналитической компетенции будущего IT-специалиста с использованием компьютерного имитационного моделирования профессиональных задач // Научные труды SWorld. 2007. Т. 14. № 3. С. 16а-18.

6. Курзаева Л.В., Чусавитина Г.Н. К вопросу о формировании требований к компетенциям личности в области информационной безопасности в системе высшего профессионального образования // Фундаментальные исследования. 2013. № 8 (часть 5). С. 1203-1207; Режим доступа: www.rae.ru/fs/?section=content&op=show_article&article_id=10001321

7. Курзаева Л.В. Ситуационные тесты в системе оценки результатов обучения личности (на примере IT-направлений подготовки) /Л.В. Курзаева //Научные труды SWorld. - 2013. - Т. 16. - № 3.- С. 61-63.

8. Курзаева Л.В. Тестовая оценка результатов обучения личности в парадигме компетентностного подхода (на примере IT- направлений подготовки) // Современные информационные технологии и ИТ-образование. - 2013. - № 9. - С. 151-156.

9. Лактионова Ю.С. Информатика : учебное пособие / Ю. С. Лактионова; М-во образования и науки Российской Федерации, Федеральное гос. образовательное учреждение высш. проф. образования ГОУ ВПО "Магнитогорский гос. ун-т". Магнитогорск, 2010. – 180 с.

10. Лактионова Ю.С. Модель развивающего обучения учащихся старших классов средней общеобразовательной школы в процессе

изучения «Информатики и ИКТ» / Сибирский педагогический журнал. 2010. № 7. С. 278-284.

11. Лактионова, Ю.С. Активизация учебной деятельности учащихся посредством дидактических игр//Информатика и образование.- Москва, 2010.-№5.-с.65-67.

12. Лактионова Ю.С. Структурные элементы дидактической системы развивающего обучения информатике // Научные труды SWorld. 2009. Т. 18. № 4. С. 33-35.

13. Лактионова Ю.С., Брябрина Л.С. Использование игровых технологий в процессе обучения школьников информатике // Научные труды SWorld. 2013. Т. 16. № 3. С. 43-46.

14. Чернова Е.В., Доколин А.С. Разработка практического курса по изучению продукта «Контур информационной безопасности» // Новые информационные технологии в образовании : материалы Всероссийской заочной электронной науч.-практ. конференции, 15-16 нояб. 2012 г. / Сев.-Вост. гос. ун-т ; [отв. ред.Т.А. Брачун]. – Магадан : СВГУ, 2013. – 128 с. : ил. – с. 80-82.

15. Чернова Е.В. Информационная безопасность для гуманитариев: учебник для студентов вузов / Е.В. Чернова. – Магнитогорск: Изд-во Магнитогорск.гос. техн. ун-та им. Г.И. Носова, 2016. – 280 с.

16. Чусавитина Г.Н. Подготовка будущих учителей к обеспечению информационной безопасности: монография /Г.Н. Чусавитина, Л.З. Давлеткиреева, Л. В. Курзаева, М.О. Чусавитин. -Магнитогорск: МаГУ, 2013 -188 с

Сравнительный анализ информационных систем для автоматизации деятельности редакции научного журнала

*Окжос К.М., Шабалина М.И.
ФГБОУ ВО «Магнитогорский технический государственный
университет им. Г.И. Носова»*

Аннотация. Выполнен обзор рынка свободно-распространяемых и проприетарных систем управления электронными научными журналами. Выделены критерии сравнения таких систем и на их основе выполнен сравнительный анализ. Сделан вывод о том, что универсальной модели системы управления электронным журналом с описанием конкретных требований и сервисов не существует.

Ключевые слова: научный журнал, информационная система (ИС), автоматизация, системы управления, анализ.

Введение. Развитие информационных технологий в целом и веб-технологий, в частности, послужило стимулом для переориентации всех

типов коммуникации, в том числе научной, в сторону виртуализации. Как известно, в информационном обществе ценность информации и нематериальных ресурсов становится все более осязаемой и подчас существенно более весомой, чем материальные активы. Поэтому актуальным стал вопрос создания информационной системы (ИС) для научного журнала. Основной целью создания ИС является сокращение временных затрат редактора на предиздательскую подготовку и отбор рукописей, а также управление процессами взаимодействия авторов и редакторов, рецензентов и секретарей в процессе подготовки статей. Внедрение ИС приводит к повышению качества редакционно-издательского процесса.

Рынок электронных научных журналов России в ближайшие годы должен показать динамику, подобную развитию рынка электронных книг [1]. По предварительной оценке объем рынка электронных книг в России по итогам 2015 года увеличился вдвое по сравнению с 2014 годом и составил 5% от оборота рынка печатной книги [2]. Поэтому осуществляется перевод процессов издания научных журналов в электронную форму. Такой переход необходим не только на этапах верстки выпусков журналов и публикации научных статей, но и на этапах их рецензирования.

Наблюдается тенденция создания ИС частными компаниями для распространения как научных, так и популярных журналов. Все ведущие издательства научной литературы используют ИС для управления распределенными редакциями. Большинство существующих в России ИС используются для создания электронных версий печатных научных журналов. Перевод редакционных процессов в электронную форму и размещение журналов в Сети нацелен не только на облегчение и удешевление работ по изданию научных журналов, но и на расширение целевой аудитории, повышение доступности журналов для научного сообщества. Следовательно, современный научный журнал помимо публикаций должен предоставлять с помощью своего веб-портала ряд дополнительных сервисов.

Периодические издания имеют сайты с электронными версиями опубликованных материалов или аннотациями статей. Эти сайты поддерживаются автономной системой управления, обеспечивающей навигацию по контенту, либо являются частью информационной системы (например, университета в целом). За последние годы было создано множество систем для автоматизации процесса подготовки научных изданий.

Свободно распространяемые системы для организации научного журнала

Наиболее распространенной бесплатной системой является Open Journal Systems (OJS). Она настраивается как облачный программный

комплекс, но может развертываться и управляться локально. Все бизнес-процессы системы настраиваются редакторами издания. OJS предоставляет специальный инструментарий для чтения и просмотра публикаций как в pdf-, так и в html-формате, доступен ряд функций для работы с библиографией, метаданными и др. Система представляет собой единую платформу для управления электронными журналами, поддерживающую широкий спектр бизнес-моделей и настроек предоставления доступа. Систему можно использовать как единую платформу для управления всеми периодическими ресурсами организации, поскольку размещаемые журналы управляются абсолютно независимо, и при этом настройки одного из них никак не влияют на работу другого. Система платформенезависима, то есть может быть установлена под любой операционной системой (ОС). В ее основе лежат свободно распространяемые PHP и Apache, а также СУБД (MySQL, PostgreSQL). Процесс установки является стандартным для систем управления сайтом, и в ее дистрибутив заложена поддержка русского языка [3].

Система распространяется по лицензии GNU/GPL. Проект постоянно развивается, выходят новые версии системы, доступна стабильная полная версия для самостоятельной установки. Она имеет хорошо документированную модульную архитектуру, что позволяет применять имеющиеся и разработать собственные функции. Система OJS имеет ролевую модель пользователей с разными правами доступа и многоступенчатый процесс публикации ресурсов, который поддерживает все стадии жизненного цикла статьи от первоначальной загрузки ее авторской редакции до размещения в интернете окончательного варианта.

ePublishing Toolkit – издательский набор инструментов, разрабатываемый обществом Max Planck Society для управления электронными журналами научного онлайн-издательства Living Reviews. Система состоит из компонент, которые могут работать независимо. Каждый компонент содержит набор функций для работы с отдельным классом объектов системы. Базовые функции, требуемые во многих компонентах, выполнены в виде библиотек. Она имеет ролевую модель пользователей с разными правами доступа и многоступенчатый процесс публикации ресурсов, который поддерживает все стадии жизненного цикла статьи от первоначальной загрузки черновика до размещения итогового варианта в интернете. Система так же является кроссплатформенной, однако для работы требуются установка Python.

GAPWorks – электронная издательская система, разработанная немецким академическим издательством (German Academic Publishers). Система реализована с использованием PHP и СУБД PostgreSQL. Она обеспечивает процесс рецензирования, функции управления

пользователями, имеет настраиваемый набор шаблонов. Сведений о развитии системы с 2006 года нет

DigitalPublishing System (DPubS) спроектирована с учетом обеспечения сохранности информационных ресурсов и отказоустойчивости. Она может работать с издательским программным обеспечением и информационными хранилищами. Система состоит из взаимосвязанных сервисов с модульной архитектурой: модуля редакционного сервиса, сервиса индексирования, поискового медиатора, модуля обратной связи, сервисов подписки, пользовательского интерфейса и администрирования.

Редакционный сервис, входящий в состав системы, обеспечивает первоначальную загрузку статей и передачу их рецензентам, дальнейшую подготовку и публикацию выпусков журналов и финальную их загрузку в хранилище DPubS. Также реализована ролевая модель пользователей с разными правами доступа.

Ее установка требует учета особенностей архитектуры и внутренних взаимосвязей элементов системы. Отсутствие обновлений с 2008 года и документации усложняют установку и внедрение системы.

Ambra Publishing System (Ambra) – система для электронного издательства, разработанная некоммерческой организацией Toraz. Система позволяет пользователям оценивать, аннотировать и комментировать публикации, что дает возможность сообществу авторов и читателей оперативно обмениваться новыми научными идеями. Ее особенностью является использование нереляционной СУБД. В системе отсутствуют специальные роли для редакторов и рецензентов. Система кроссплатформенна, однако дистрибутив не содержит мастера-установщика, поэтому ее сложно установить. Последнее обновление системы датировано 2009 годом.

Платные системы для организации научного журнала

Рассмотрим платные системы для организации электронной формы научного журнала.

Автоматизированная система «Спринт» подробно описана в [3]. Она работает на базе единой программной платформы и позволяет создавать научные электронные журналы различной тематики. В программе реализованы возможности создания, изменения и формирования всех необходимых для функционирования журнала элементов: поиск статей, назначение рецензентов и генерация pdf-версий статей журнала. Существенным минусом этой программы является частичная автоматизация процесса рецензирования и отсутствие личных кабинетов для некоторых групп пользователей научного журнала.

Платформа для электронных научно-образовательных журналов «е-НОЖ» представляет необходимый инструментарий для создания электронного издательства. К основным ее функциям относятся:

создание научной публикации, проверка, рецензирование и включение публикации в выпуск электронного журнала; поддержка версионности публикаций; разграничение прав пользователей: авторы, редакторы, главные редакторы, рецензенты [4]. Однако отсутствие системы поддержки принятия решения для оптимизации процесса рецензирования и пользователя с правами секретаря являются недостатками рассмотренной платформы.

Программный комплекс «Электронный научный журнал» (ЭНЖ) создан коллективом авторов из «ПетрГУ», на базе этого комплекса создано более пяти научных изданий. Он позволяет автоматизировать полный цикл создания выпусков журнала. В нем реализованы возможности создания, изменения и формирования всех необходимых для функционирования журнала элементов [5].

«Журнальная интернет-система импорта и экспорта научных статей» (ЖИС) позволяет осуществлять представление научных журналов в сети Интернет. Система предназначена для сотрудников издательств и издательских домов и способствует повышению индекса цитирования путем публикации материалов научных журналов в сети Интернет. В частности, она позволяет: подавать и моделировать статьи в электронном виде; публиковать научные статьи в сети Интернет; формировать электронный архив; осуществлять многокритериальный поиск по сформированной базе [5].

Рассмотрим критерии, выделенные для сравнения систем управления электронными научными журналами:

- **ролевая модель** – наличие правила разграничения доступа для пользователей системы. Наличие ролей в системе управления научным журналом помогает организовать процессы предиздательской обработки поступающих в редакцию материалов;
- **полный цикл подготовки статей** – в системе реализованы все инструменты для организации процессов предиздательской обработки поступающих материалов и рецензирования;
- **мультязычность** – это возможность функционирования сайта на нескольких языках;
- **модульность** – свойство системы, связанное с возможностью ее декомпозиции на ряд внутренне связанных между собой модулей. Модульная организация системы управления научным журналом облегчит процессы реорганизации, связанные с наращиванием ее функционала;
- **кроссплатформенность** – возможность программного обеспечения работать более чем на одной аппаратной платформе и/или операционной системе. Таким образом, не нужно иметь

несколько версий систем управления электронными научными журналами для разных типов архитектур серверов;

- **бесплатность** – возможность использования программного обеспечения без дополнительной платы.

Результаты сравнения рассмотренных выше систем представлены в таблице 1:

- 1) OJS;
- 2) ePubTK;
- 3) DPubS;
- 4) GAPWorks;
- 5) Ambra;
- 6) eJournal;
- 7) Спринт;
- 8) eНОЖ;
- 9) ЭНЖ;
- 10) ЖИС;
- 11) EIPub.

При сравнении систем применялась следующая система оценок: если рассматриваемая система полностью удовлетворяет критерию, то оцениваем его на «+», если удовлетворяет частично оцениваем на «±», в противном случае присваиваем «-».

Таблица 1

Сравнительная таблица систем управления журналами

Критерий	Системы										
	1	2	3	4	5	6	7	8	9	10	11
Ролевая модель	+	+	+	+	±	+	+	+	+	+	+
Полный цикл подготовки статей	+	+	+	+	±	±	+	±	+	+	+
Мультиязычность	+	-	-	-	-	-	-	-	-	-	-
Модульность	+	+	+	+	+	+	±	±	±	±	±
Кроссплатформенность	+	+	+	+	+	+	-	-	-	-	-
Бесплатность	+	+	+	+	+	+	-	-	-	-	-

Выводы

Анализ, систематизация и обобщение научно-технической информации по теме исследования, позволили определить актуальность исследуемой темы и направления для её развития.

Большинство рассмотренных информационных систем, разрабатывались для обеспечения функционирования конкретных электронных изданий. Это привело к существенным различиям архитектуре систем и функциональных возможностях. Следовательно, не существует универсальной модели системы управления электронным журналом с описанием конкретных требований и сервисов. На текущий момент времени большинство проектов не получило развития, исключением является OJS. Так же стоит отметить, что большинство систем не являются мультиязычными, что затруднит их использование авторами из РФ. Лицензия на большинство платных систем стоит от 20 до 50 тысяч в год, поэтому их использование для развивающихся научных изданий не рентабельно. У большинства систем реализованы механизмы ролевых политик и модульности, однако для применения их в определенном журнале требуется значительная модификация.

Список используемых источников

1. Гусев А.Л. Анализ рынка услуг издательских платформ по управлению деятельностью распределенных коллегий электронных изданий // Информационные технологии. – 2013. – № 04/1 (123). – С. 82–86
2. Cnews [Электронный ресурс]: Рынок электронных книг в России. – Электрон. текстовые данные (14035 bytes). Режим доступа: <http://www.cnews.ru> (дата обращения: 02.03.2016).
3. Елизаров А. М. Свободно распространяемые системы управления электронными журналами и технологии электронных библиотек / А. М. Елизаров, Д. С. Зуев, Е. К. Липачёв // Электронные библиотеки: перспективные методы и технологии, электронные коллекции. – 2013. – С. 105-115.
4. Свид. 2015611117 Российская Федерация, Автоматизированная система для создания научных электронных журналов «Спринт» / О.Ю. Насадкина, А.Г. Марахтанов, А.А. Кухарская, Е.В. Голубев. – № 2014661940; заявл. 24.11.2014; опубл. 2015.02.20.
5. Свид. 2013614347 Российская Федерация, Платформа для электронных научно-образовательных журналов «е-НОЖ» / Д.В. Фомин-Нилов, И.А. Тарханов. – № 2013612278; заявл. 19.03.2013; опубл. 2013.06.20.
6. Свид. 2013613707 Российская Федерация, Программный комплекс для ЭВМ «Электронный научный журнал» / В.С. Сюнёв, А.Г. Марахтанов, В.П. Банкет, А.А. Кухарская, А.В. Коросов. – № 2013611739; заявл. 20.02.2013; опубл. 2013.06.20.
7. Свид. 2013615525 Российская Федерация, Журнальная интернет–система импорта и экспорта научных статей / А.А. Бондарев, Д.С. Егоров. – № 2013613093; заявл. 16.04.2013; опубл. 2013.09.20.

8. Логунова О.С. Методика исследования предметной области на основе теоретико-множественного анализа / О.С. Логунова, Е.А. Ильина // Математическое и программное обеспечение систем в промышленной и социальной сферах. – 2012. – №2. – С. 281-291.

9. Логунова О.С. Структуризация лексикографической информации при разработке программного обеспечения / О.С. Логунова, Е.А. Ильина // Математическое и программное обеспечение систем в промышленной и социальной сферах. – 2014. – №1. – С. 87-91.

10. Логунова О.С. Наукометрические показатели в деятельности ученого / О.С. Логунова, Д.Я. Арефьева // Ab ovo ... (С самого начала ...). – 2014. – С. 102-103.

11. Окжос К.М. Описание модели базы данных для информационной среды научного журнала «Ab ovo...» (С самого начала...) // Сборник научных трудов SWorld. – 2015. – Т. 4. – № 1 (38). – С. 64-67.

12. Нургалина Р.Г. Принятие решений при измерении уровня рефлексии в системе дистанционного обучения / Р.Г.Нургалина, Е.А. Ильина // Математическое и программное обеспечение систем в промышленной и социальной сферах. – 2012. – №2. – С. 250-256.

13. Логунова О.С. Индексный анализ управления публикационной активностью научно-педагогических работников вуза и его результаты / О.С. Логунова, Д.Я. Арефьева, Е.А. Ильина // Актуальные проблемы современной науки, техники и образования. – 2015. – Т. 2. – № 1. – С. 198-204.

14. Окжос К.М. Об информационной среде научных журналов / К.М. Окжос, Е.А. Кустыбаева, Е.А. Ильина // Сборник научных трудов SWorld. – 2014. – Т.8, № 1. – С. 25-29.

15. Ильина Е.А. Структура информационной среды научного журнала / Е.А. Ильина, К.М. Окжос // Ab ovo ... (С самого начала ...). – 2014. – С. 62-64.

16. Кустыбаева Е.А. Разработка системы принятия решения для рецензирования статей // Е.А. Кустыбаева, К.М. Окжос, Е.А. Ильина / Научные труды SWorld. – 2014. – Т. 8. – № 2. – С. 29-31.

17. Окжос К.М. К вопросу об информационном и программном обеспечении в издательской деятельности // Наука и образование в XXI веке. – 2014. – С. 99–101.

18. Логунова О.С. Система поддержки принятия решения для оценки качества статей научного журнала / О.С. Логунова, Е.А. Ильина, К.М. Окжос // Фундаментальные исследования. – 2016. – № 2 (3). – С. 492–497

19. Логунова О.С. Система оценки качества статей научного журнала / О.С. Логунова, Е.А. Ильина, К.М. Окжос // Математическое и

программное обеспечение систем в промышленной и социальной сферах. – 2015. – № 1 (7). – С. 56–57.

20. Логунова О.С. Динамика показателей публикационной активности профессорско-преподавательского состава магнитогорского государственного технического университета им. Г.И. Носова / О.С. Логунова, Л.Г. Егорова, В.В. Королева // Вестник Магнитогорского государственного технического университета им. Г.И. Носова. – 2015. – № 3 (51). – С. 101–112.

21. Логунова О.С. Результаты анализа публикационной активности профессорско-преподавательского состава ФГБОУ ВПО «Магнитогорский государственный технический университет им. Г.И. Носова» / О.С. Логунова, А.В. Леднов, В.В. Королева // Вестник Магнитогорского государственного технического университета им. Г.И. Носова. – 2014. – № 3 (47). – С. 78–87.

Интерактивное видео: инструменты и возможности применения

*Пуляевская А. М., доцент кафедры информатики и математики,
НОУ ВПО «Сибирская академия права, экономики и управления»*

*Вислова К. В.,
Евразийский лингвистический институт ФГБОУ ВО «Московский
государственный лингвистический университет»*

Аннотация. В статье представлен обзор онлайн-сервисов для создания интерактивного видео. Раскрываются функциональные возможности приложения H5p и демонстрируются примеры реализации интерактива на его базе.

Ключевые слова: интерактивное видео, онлайн-сервисы, сказки, проект, H5p, «Сказочный алфавит».

В 2010 году общество «ИкаРус-Межкультурная коммуникация и русский язык», руководителем которого является Кудрявцева Екатерина (ФРГ), провело совместно с многочисленными друзьями и партнерами Второй всемирный конкурс «Дети рисуют свой русский мир» на тему «Волшебный мир русской сказки». Полученные работы вдохновили организаторов, членов жюри и педагогов русских центров ФРГ, Италии, Эстонии, Приднестровья на разработку проекта «Сказочный алфавит» [1]. Этот проект был завершён весной 2012 года.

В настоящее время проект выходит на новый уровень — электронный. В разработке принимают участие студенты профиля «Теоретическая и прикладная лингвистика» направления подготовки Лингвистика Московского Государственного Лингвистического

Университета Евразийского Лингвистического Института Филиала МГЛУ в г. Иркутске.

Все сказки из серии «Сказочный алфавит» представляют собой электронные книги, которые содержат:

- текст сказки;
- словарь устаревших слов;
- аудио (аудио-сказки) и видео (мультфильмы, песни, видео-уроки) сопровождение;
- задания и упражнения, разработанные авторами проекта «Сказочный алфавит»;
- интерактивные упражнения;
- раскраски и др.

Мультфильмы к сказкам представляются в формате интерактивного видео. Интеракция (картинки, текст, тесты) возникает на экране в ходе просмотра видео обучающимися.

В настоящее время для создания такого видео не требуются навыки программирования, а сам процесс не занимает много времени. В таблице 1 приводится описание функциональных возможностей популярных в Интернете бесплатных и условно-бесплатных онлайн-сервисов.

Таблица 1

Обзор функциональных возможностей сервисов

Сервис	Интерактив					Дополнительные возможности	Комментарий
	Текст	Гиперссылки	Картинки	Викторины	Опрос		
Zaption	+	+	+	+	+	Рисование, приобретение платного аккаунта.	Возможна загрузка собственных видео.
Edpuzzle	+	+	+	+	-	Нарезка видео, добавление голосовых комментариев. Доступна статистика выполнения заданий.	Возможна загрузка собственных видео.

Сервис	Интерактив					Дополнительные возможности	Комментарий
	Текст	Гиперссылки	Картинки	Викторины	Опрос		
Playposit	-	+	-	+	-	Приобретение платного аккаунта. Вставка вопросов, требующих письменного ответа. Нарезка видео. Доступна статистка.	Поиск видео по заданной тематике на хостингах.
Thinglink	+	+	-	-	-	Приобретение платного аккаунта.	Загрузка видео исключительно с хостингов.
Wirewax	+	-	+	-	-	Добавление видео.	Возможна загрузка собственного видео.
Wideo	+	+	+	-	-	Добавление звуков. Бесплатные ролики до 90 секунд. Приобретение платного аккаунта.	Сервис для создания видеопрезентаций. Выбор либо готовых шаблонов, либо чистого листа.
Videopath	+	+	+	-	-	Приобретение платного аккаунта.	Ролики исключительно с YouTube.
H5P	+	+	+	+	-	Разнообразие добавляемых викторин.	Возможна загрузка собственных видео.
Наруак	+	+	+	+	+	Рисование. Приобретение платного аккаунта.	Возможна загрузка собственного

Сервис	Интерактив					Дополнительные возможности	Комментарий
	Текст	Гиперссылки	Картинки	Викторины	Опрос		
							видео.

Все рассмотренные сервисы нерусифицированы. При сопоставлении возможностей сервис H5P был признан нами самым простым и функциональным. Опишем процесс создания интерактивного видео [2].

На рисунке 1 представлены три вкладки для последовательного создания интерактивного видео: загрузка видео (Uploadvideo), добавление интеракций (Addinteractions) и создание финального теста (Summarytask).



Рис. 1. Вкладки шаблона для создания интерактивного видео

Рассмотрим подробнее каждый из этапов.

1 этап - Uploadvideo

Видео может быть загружено как с компьютера (в формате mp4), так и представлено ссылкой на размещение в видео-хостинге YouTube. Для того чтобы видео было доступно для пользователя с любым браузером, разработчики предлагают дополнительно добавлять видео в формате webm. Для соблюдения авторских прав необходимо представить следующую информацию:

- название видео;
- автор;
- год создания;
- ссылку на источник.

2 этап - «Addinteractions»

На рисунке 2 представлена панель инструментов для добавления интерактива (слева направо):

1. Label – краткие текстовые пояснения.
2. Text – полнотекстовую информацию.
3. Table – таблицы.
4. Link – гиперссылки.

5. Image – изображения.
6. Statements – викторина на выбор единственно правильного утверждения из нескольких предложенных.
7. Singlechoiceset – тест с выбором единственного правильного ответа из нескольких предложенных.
8. Multiplechoice – тест с множественным выбором правильного ответа из нескольких предложенных.
9. Fillintheblanks – упражнение «Заполни пропуски».
10. Markthewords – упражнение «Выдели слова».
11. Dragtext – упражнение «Расставить правильно».
12. Gotoquestion – викторина с переходом, т.е. в зависимости от выбора ответа, ученик переходит в определённое место в хронометраже видео.

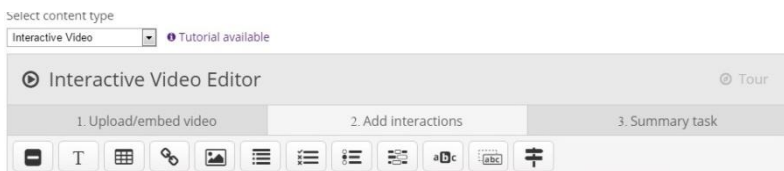


Рис. 2. Панель инструментов интерактива для видео

На рисунке 3 представлен шаблон для ввода текстовой информации с комментариями.

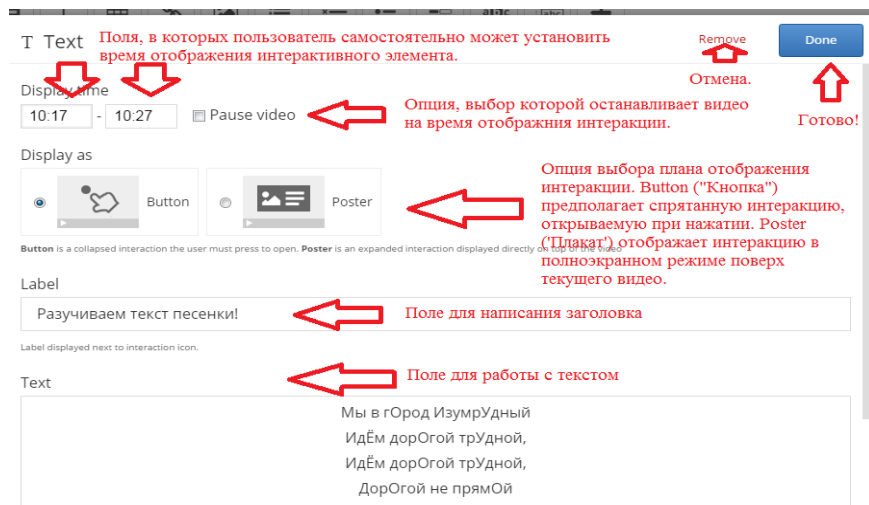


Рис.3. Вид окна шаблона для добавления текстовых сообщений с комментариями

На рисунке 4 представлена стандартная панель инструментов при работе с текстом: вид шрифта (полужирный, курсив, зачёркнутый, сброс форматирования), тип выравнивания текста (по левому/правому краю, центру), автоматическое маркирование/нумерация списка, вставка/удаление гиперссылок.



Рис. 4. Панель инструментов для работы с текстом

Пример реализации представлен на рисунке 5.



Рис. 5. Пример представления текстового комментария на видео

Как видно на рисунке 5 на экране видео расположена иконка в виде синей кнопки с плюсом, справа от которой располагается введённый автором заголовок. Синяя точка появится и на временной шкале видео, это позволит определить наличие и время демонстрации интерактивных элементов. Кнопку на экране можно переместить на любое удобное место способом drag-and-drop.

Аналогично размещение интеракции в виде картинки.

Рассмотрим создание теста с выбором одного правильного ответа из нескольких предложенных. Для его создания на панели инструментов пользователь выбирает «Clickanddragtoplacemultiplechoice». На рисунке 6 представлен шаблон для добавления теста с комментариями.

Третий этап работы с интерактивным видео — создание итогового теста, так называемого Summary. Разработчики предлагают в качестве итогового задания выбор единственно правильного утверждения из нескольких. На рисунке 7 представлен шаблон Summary с комментариями.

Конструктор предлагает также возможность создания и использования закладок, которые могут отображать начало новой темы, интерактивное задание или важное событие, происходящее на видео. Все закладки отображаются в соответственном меню закладок «Bookmark». На временной шкале видео каждая закладка выделена вертикальной серой чертой. При необходимости обучающийся может непосредственно проходить на нужные места в видео, нажимая на эти серые пометы. Для

добавления закладки достаточно выбрать место в хронометраже видео и нажать «Addbookmark».

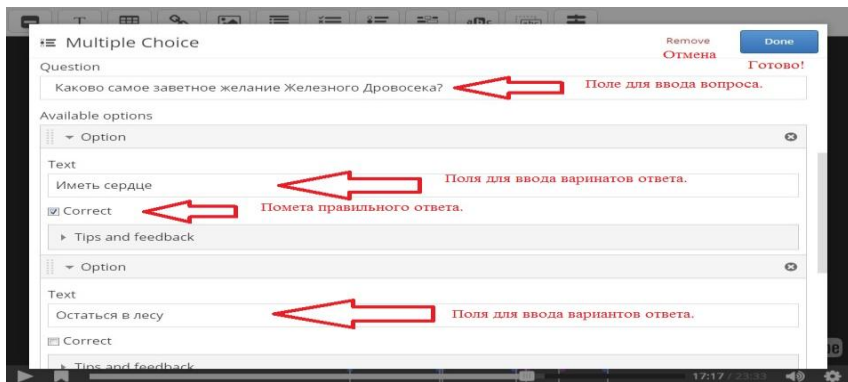


Рис. 6. Шаблон теста с комментариями

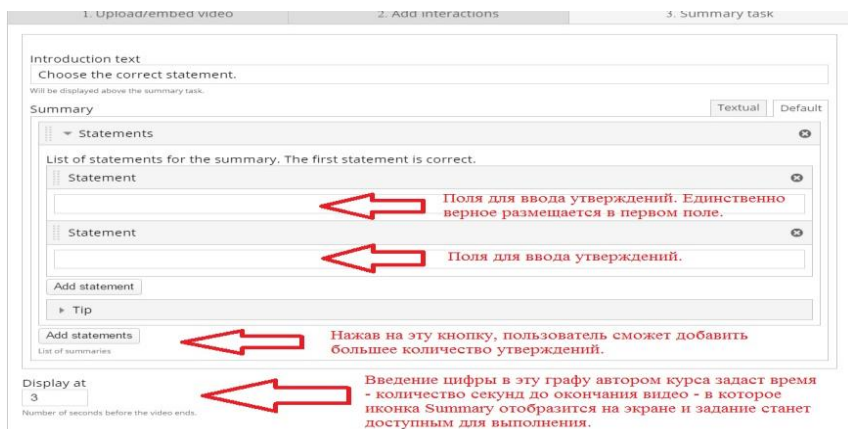


Рис. 7. Шаблон Summary с комментариями

Платформа H5P совместима с Drupal, WordPress, Moodle. Интеракция, созданная с помощью H5P, доступна как на компьютере, так и на мобильных устройствах (смартфоны, планшеты).

Например, для создания обучающего видео к сказке «Волшебник Изумрудного Города» были использованы первые пять серий одноименного мультфильма, созданного творческим объединением «Экран» в 1973-1974 годах:

1 серия «Элли в Волшебной Стране» (h5p.org/node/11449) — интерактивные элементы: 4 текста песен героев, 3 викторины.

2 серия «Дорога из Жёлтого Кирпича» (h5p.org/node/11814) — интерактивные элементы: 2 текста песен героев, 2 викторины, задание.

3 серия «Изумрудный город» (h5p.org/node/12317) – интерактивные элементы: задание.

4 серия «Королевство Бастинды» (h5p.org/node/12519) - интерактивные элементы: текст песни.

5 серия «Разоблачение Великого и Ужасного» (h5p.org/node/12529) - интерактивные элементы: 2 текста песен, словарь, гиперссылки на продолжение мультфильма.

В заключении отметим, что в перспективе подготовка комплекса интерактивных учебных ресурсов и создание дистанционного курса на базе платформы H5P в рамках проекта «Сказочный алфавит» для педагогов, родителей и детей.

Список использованных источников

1. Кудрявцева Е.Л. Игры как способ развития речи и национального самосознания у билингвов (на примере подвижных игр для билингвов и игр по мотивам русских сказок) (тезисы) [Электронный ресурс] / Ассоциация русскоязычных образовательных центров POLYLOG e.V. [сайт]. Режим доступа: <http://www.bilingual-online.net> (дата обращения: 8.05.2016)

2. Tutorials for authors // H5P Documentation. Режим доступа: <https://h5p.org> (дата обращения: 8.05.16)

Индивидуальная траектория студента как часть автоматизированной системы составления расписания в вузе

Рабовская М. Я., к.ф.-м.н.

Махнева В.В.,

ФГБОУ ВО «Уральский федеральный университет им. Б.Н.

Ельцина»

Аннотация. В статье приводится описание разработки модуля автоматизированного составления расписания, выполненного с помощью JavaPlatformSE 7. Представлены постановка задачи, проектные решения по информационному и программному обеспечению модуля.

Ключевые слова: автоматизация составления расписания, индивидуальная образовательная траектория, программный модуль, Java

Введение

В соответствии с дорожной картой первоочередных мероприятий по модернизации образовательной деятельности ФГАОУ ВПО Уральский федеральный университет имени первого Президента России Б.Н.

Ельцина (УрФУ) переходит к реализации образовательных программ высшего образования (ОПВО), обеспечивающих возможность реализовать обучающимися выбранные ими образовательные траектории.

Образовательная программа высшего образования (ОПВО) - комплекс основных характеристик образования (объем, содержание, планируемые результаты), организационно-педагогических условий и форм аттестации, который представлен в виде основной характеристики образовательной программы, учебного плана, календарного учебного графика, рабочих программ дисциплин (модулей), а также оценочных и методических материалов.

Индивидуальная образовательная траектория - это персональный путь освоения ОПВО, который состоит из модулей ОПВО. Индивидуальная образовательная траектория представлена индивидуальным учебным планом обучающегося, состоящим из обязательных модулей (инвариантных), а также выбранных обучающимся из предложенного набора модулей. Индивидуальные учебные планы проектируются на основе заданных основной характеристикой ОПВО правил, представляют собой версию единого учебного плана ОПВО.

Модуль – это компонент ОПВО, включающий дисциплины(у), а также, по необходимости, междисциплинарные проекты, которые обеспечивают формирование предусмотренного для данного модуля набора результатов обучения.

В университете различают следующие **типы модулей**, которые включены в ОПВО и формируют индивидуальную образовательную траекторию освоения программы и соответствующий индивидуальный учебный план обучающегося:

1. **Унифицированные модули** - это модули, отвечающие за освоение общепрофессиональных и/или общекультурных компетенции, совпадающих для нескольких направлений/специальностей подготовки. Они в обязательном порядке включаются в индивидуальный учебный план обучающихся по всем траекториям ОПВО, составляя его инвариантную часть.

2. **Обязательные профессиональные модули** - это модули, отвечающие за освоение общепрофессиональных и профессиональных компетенции (ОПК и ПК) по направлению/специальности подготовки. Они в обязательном порядке включаются в индивидуальный учебный план обучающихся, составляя его инвариантную часть.

3. **Модули по выбору** – это модули, которые обеспечивают индивидуализацию образовательной траектории и после выбора обучающимся включаются в его индивидуальный учебный план.

Переход на индивидуальные образовательные траектории, выделение унифицированных модулей для широкого применения в рамках различных образовательных программ, введение в состав образовательных программ майноров обеспечивают расширение образовательных возможностей студентов и повышение их уровня ответственности за результаты обучения.

Реализация нового подхода невозможна без поддержки со стороны информационных технологий. В данной статье рассматривается процесс автоматизированного составления расписания.

Общие принципы автоматизированного составления расписания с учетом индивидуальной траектории студента

Информационные технологии могут эффективно использоваться в создании и реализации новых стратегий образовательной политики, и даже стать решающим фактором в изменении стратегии. Особый подход – создание динамической организационной среды, которая позволяет автоматизировать существующие бизнес-процессы, основываясь на комплексных знаниях.

Для реализации данного подхода была разработана автоматизированная система составления индивидуального расписания студента. В основу функционала этой системы были положены следующие процессы:

- совмещение двух режимов составления расписания: по академическим группам и по дисциплинам модулей учебных планов;
- формирование потоков: разрешено объединять следующие объекты системы - академические группы, подгруппы;
- проектирование экранных форм планового и ежедневного расписания, расписания аудиторий и преподавателей по дисциплинам модулей;
- проектирование печатных форм расписания дисциплин модулей, преподавателей и аудиторий в различных форматах;
- реализация интеграционных модулей.

Введем основные термины, используемые для автоматизированного составления расписания.

Укрупненная группа – группа, обучающаяся на одной версии образовательной программы с учетом атрибутов ОП:

- начальный учебный год;
- курс;
- направление (специальность);
- уровень обучения;
- профиль / магистерская программа;
- форма освоения (очная, очно-заочная, заочная);
- технология освоения (традиционная, с применением электронного обучения, дистанционных образовательных технологий);

- условия освоения (полный срок, ускоренная программа);
- подразделение;
- вид возмещения затрат.

Подгруппа – сущность процессов, связанных с составлением расписания по видам нагрузки и в конечном итоге закрепление студентов за конкретным мероприятием расписания.

Механизм подгрупп универсален по отношению к задачам по составлению расписания:

- распределение студентов с учетом ограничения ресурсов;
- распределение студентов с учетом их пожеланий по условиям проведения занятий (время, преподаватель);
- распределение студентов с учетом условий организации процесса обучения по отдельным дисциплинам, например, секции или подгруппы по языковым компетенциям.

Система «Расписание» является системой с сильно интегрированной архитектурой. Через настраиваемый модуль импорта/экспорта проходят большие потоки данных, диаграмма которых представлена на рисунке 1Рис. 9: организационная структура (институты, департаменты), заявки на учебную нагрузку, лимиты на модули, преподаватели, траектории, заявления студентов о выборе индивидуальных траекторий; подгруппы на дисциплины модулей по видам нагрузки, кол-во человек в подгруппе / лимиты на модули, расписание занятий студентов и преподавателей.

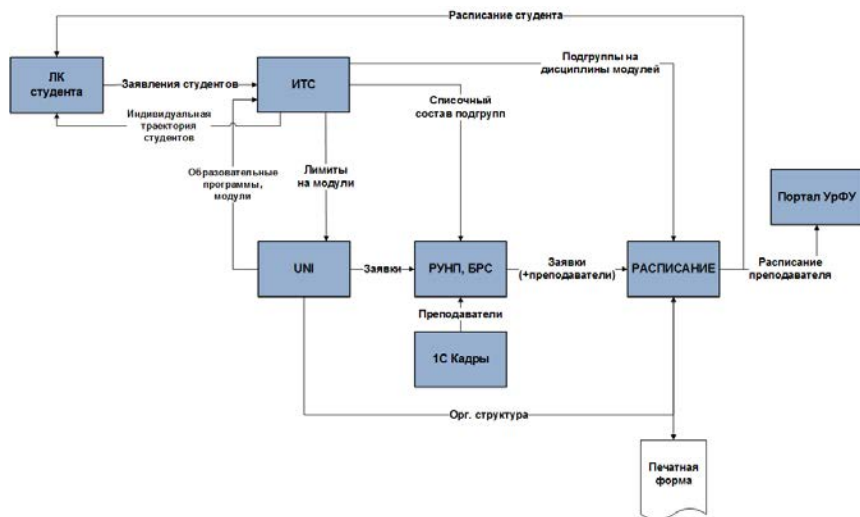


Рис. 9. Диаграмма потоков данных

Названия подгрупп формируются согласно внутренним правилам университета, а именно, автоматическим добавлением ряда параметров (название дисциплины, сокращенное название вида нагрузки, порядковый номер подгруппы) к названию укрупненной группы, например, СТ-140001/Математика/п1.

Система предназначена для составления расписания дисциплин модулей с учетом индивидуальной траектории студента (ИТС), в ходе которого реализуются следующие проверки: по занятости преподавателей и аудиторий, по виду нагрузки внутри дисциплины. Также предусмотрена зависимость подгрупп разных видов нагрузок внутри одной дисциплины и реализована сквозная нумерация подгрупп. Схема ее иерархической зависимости представлена на рисунке 2.

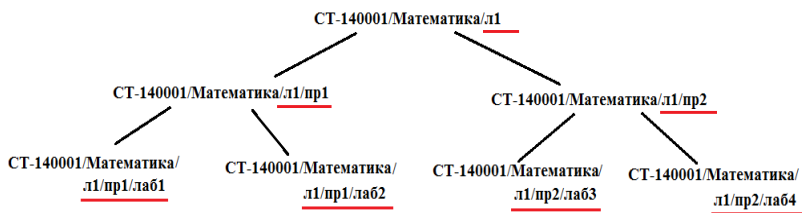


Рис. 10. Иерархическая структура подгрупп (фрагмент)

Интерфейсное решение позволяет визуализировать составление расписания, оперируя не академическими группами, а дисциплинами модулей учебных планов. Экранная форма расписания дисциплин модулей представлена на рисунке 3.

Дисциплины
 Обновить
 Печать
 Создать ежедневное расписание

▼ Понедельник			
	Основы менеджмента и маркетинга в строительстве Менеджмент и управление проектами в строительстве	Основы системного анализа Информационные технологии проектирования инженерных систем	Планировка, застройка и реконструкция населенных мест Градостроительное проектирование
1 пара 08:30 - 10:00			
2 пара 10:15 - 11:45	СТ-240037/л1/лн1 практические занятия Усков А.Ю. С211 (Еженедельно)		СТ-240037/л1 лекции Пенцев Е.А. С204
3 пара 12:00 - 13:30	практические занятия Усков А.Ю. С211		СТ-240037/л1/лн1 практические занятия Пенцев Е.А. С204

Рис. 11. Экранная форма планового расписания дисциплин модулей

Система «Расписание» организована по принципу трехзвенной архитектуры (рисунок 4): клиент, сервер приложений и сервер баз данных.

Сервер приложений предоставляет программный интерфейс для доступа к данным. Основной протокол обмена данными – HTTP.

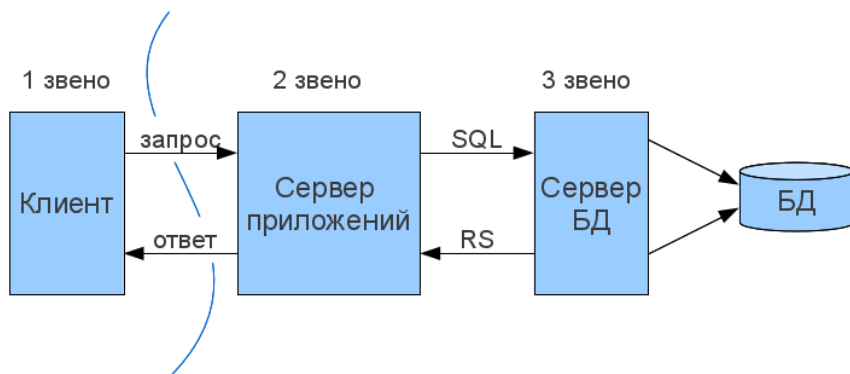


Рис. 12. Принцип трехзвенной архитектуры

В качестве основы системы, предоставляющей среду выполнения и все механизмы низкоуровневого взаимодействия с аппаратным обеспечением, выступает программная платформа JavaPlatformSE 7.

Модуль «Расписание» реализован на открытых программных компонентах как в серверной части, так и клиентской, с использованием сторонних библиотек.

Заключение

Таким образом, разработанная система формирования расписания позволяет гибко составлять расписание студентов с учетом его индивидуального выбора модулей и траекторий. В дальнейшем планируется расширение функционала в части выбора студентами расписания модулей через личный кабинет.

Список использованных источников

1. Дорожная карта первоочередных мероприятий по модернизации образовательной деятельности УрФУ, утвержденная ректором УрФУ – Кокшаровым В.А.
2. Образовательная политика в части реализации образовательных программ бакалавриата, специалитета и магистратуры. Одобрена решением Ученого совета ФГАОУ ВПО «УрФУ имени первого Президента России Б.Н. Ельцина», протокол № 9 от 26.10.2015. Утверждена 10.12.2015 ректором УрФУ Кокшаровым В.А.

**Разработка электронного журнала научных статей кафедры
Вычислительной техники и инженерной кибернетики Уфимского
государственного нефтяного технического университета
для публикации в сети интернет на базе Open Journal Systems**

*Фукалов Д.С.,
ФГБОУ ВПО «Уфимский государственный нефтяной технический
университет»*

Аннотация. Разработан электронный журнал научных статей кафедры Вычислительной техники и инженерной кибернетики (ВТИК) на базе открытого программного обеспечения для организации рецензируемых научных изданий Open Journal Systems (OJS). Журнал принимает для публикации статьи, относящиеся ко всем областям прикладных, естественных и гуманитарных наук. По желанию авторов возможна публикация на английском и русском языках. Все публикуемые материалы проходят тщательный отбор и рецензируются. К рецензированию привлекаются ведущие специалисты УГНТУ и кафедры ВТИК. Статьи по мере поступления публикуются в виде электронных файлов в формате PDF и размещаются в Интернете на веб-сайте журнала: journal.vtik.net.

Ключевые слова. Электронный журнал, Open Journal Systems, открытое программное обеспечение, публикация.

Целью данного электронного журнала является повышение публикационной активности преподавателей Уфимского Государственного Нефтяного Технического Университета. Платформой для этого журнала было выбрано открытое программное обеспечение OpenJournalSystems.

OpenJournalSystems (OJS) - открытое программное обеспечение для организации рецензируемых научных изданий, разработанное некоммерческим исследовательским проектом PublicKnowledgeProject. OJS можно загрузить бесплатно и установить на свой веб-сервер, он позволяет управлять журналом и его изданиями.

Основные преимущества.

1. OJS полностью бесплатен и распространяется по лицензии GNUGeneralPublicLicense.
2. OJS содержит большое количество различных модулей.
3. OJS поддерживает более чем 30 языков.

4. OJS - это кроссплатформенное приложение, которое может быть запущено на любом веб-сервере. Оно написано на PHP и использует базу данных MySQL.

OpenJournalSystems позволяет настраивать сайт для решения различных задач, такие как подача статьи, рецензирование, редактирование, публикация, архивирование и индексирование журнала.

OJS позволяет управлять статьями в журнале с помощью редакционных этапов, которые управляются редакторами журнала:

- Очередь неназначенных статей: На этом этапе статья назначается одному или нескольким редакторам.
- Рецензирование статьи: Этап охватывает рецензирование и редакционное решение о публикации.
- Редактирование статьи: Этап включает литературное редактирование, верстку и корректуру. Статья закрепляется за определенным выпуском.
- Содержание: Статьи выстраиваются в необходимом порядке и публикуются.

Установив OpenJournal Systems на веб-сервер, можно управлять сразу несколькими журналами. У каждого журнала есть собственный URL, а также у журналов есть собственный интерфейс, который можно изменять. OJS позволяет администратору назначить редактора, который может управлять всеми аспектами журнала и сайта.

Так же OJS позволяет управляющему журнала распределять редакционные роли:

- Управляющий журнала: Настраивает журнал и распределяет редакционные роли (может при этом выступать в нескольких ролях).
- Редактор: Следит за редакционным процессом; может назначать статьи редакторам разделов, которые управляют рецензированием и редактированием статей; следит за графиком издания журнала.
- Редактор раздела: Управляет рецензированием и редактированием принятых к публикации статей.
- Литературный редактор: Работает со статьей с целью улучшить грамматику и ясность материала, отправляет автору запросы касательно возможных ошибок, обеспечивает высокое качество стиля статьи и библиографии.
- Верстальщик: Преобразует отредактированные статьи в гранки форматов HTML, PDF, и/или PS для последующей публикации.
- Корректор: Вычитывает гранки на предмет ошибок правописания и форматирования.

Необходимость в собственном журнале кафедры обусловлена тем, что администрация (руководство факультета, кафедры) на основе комплексной оценки уровня компетентности, профессионализма и педагогического мастерства, результатов деятельности преподавателя на протяжении всего его контракта делает выводы о подтверждении его квалификации в соответствии с настоящими квалификационными требованиями, предоставляет ему возможность повысить свою квалификацию, формулирует рекомендации о возможности заключения контракта на последующие сроки.

Необходимым условием получения научных званий за период работы по контракту (при условии его заключения сроком на 5 лет) является наличие определенного числа публикаций научных статей.

Ассистент (преподаватель) должен опубликовать не менее 2 научных работ и 2 учебно-методических разработок.

Старший преподаватель должен опубликовать не менее 3 научных работ и учебно-методических разработок.

Доцент должен опубликовать не менее 5 научных работ или монографию и не менее 3 учебно-методических разработок.

Профессор должен опубликовать не менее 2 учебно-методических разработок и 1 учебно-методического пособия, а также не менее 5 научных работ (статей, изобретений) или монографию.

Преподаватель должен участвовать в научно-исследовательской работе, проводимой кафедрой, в планировании, реферировании, экспериментальной работе по одному из направлений исследования, анализируя результаты и подготавливая их к публикации. Преподаватель имеет право использовать для публикации как печатные, так и электронные научные журналы.

Таблица 1

Условия получения научных званий

Степень	Научные работы	Учебно-методические разработки	Учебно-методическое пособие	Монографии
Ассистент (преподаватель)	2	2	-	-
Старший преподаватель	3	3	-	-
Доцент	5	3	-	1
Профессор	5	2	1	1

Первоначально преподаватели кафедры вычислительной техники и инженерной кибернетики (ВТИК) пользовались альтернативными электронными изданиями, например, электронным журналом издательства Notabene. Руководство кафедры ВТИК было недовольно ограниченным числом рубрик этих издательств.

Разработка электронного журнала кафедры вычислительной техники и инженерной кибернетики (ВТИК) решает проблему рубрикации.

В таблице 2 перечислены достоинства и свойства системы.

Таблица 2

Достоинства	Свойство системы
1	2
Бесплатная публикация	Размещение на домене кафедры ВТИК, позволит авторам из УГНТУ публиковаться бесплатно.
Большой список категорий	В электронном журнале можно создать большое количество рубрик и редактировать их по усмотрению руководства кафедры ВТИК.
Простой доступ к журналу	Получить доступ к электронному журналу можно с любого устройства: ПК, ноутбук, телефон и др. Нужен лишь браузер.
Доступ с любого ПК университета	Студенты и преподаватели могут получить доступ к журналу с любого ПК подключенного к интернету.
Простой и удобный доступ с домашнего ПК	Студенты и преподаватели могут получить доступ к журналу с любого ПК университета или со своего домашнего ПК подключенного к интернету.
Безопасность и конфиденциальность	Для доступа к работам авторов необходимы пользовательский идентификатор и пароль. Пользовательские данные защищены от несанкционированного доступа.

На данный момент реализована большая часть функций, требующихся для публикации статей авторами из Уфимского Государственного Нефтяного Технического Университета.

Интерфейс журнала реализован на языке гипертекста HTML; модули написаны на языке PHP, JavaScript; используется база данных MySQL.

Внешний вид электронного журнала был разработан с использованием модуля тем от Mason Publishing Group. Этот модуль полностью бесплатен и скачать его исходный код можно с Github.



ФГБОУ ВО "У
нефтяной т

**Факультет автоматизации производственных
КАФЕДРА ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ И ИНЖЕНЕРНОЙ КИБЕРНЕТИКИ**

[Главная](#) / [Вычислительная Техника и Инженерная Кибернетика](#)

Вычислительная Техника и Инженерная Кибернетика

Уважаемые коллеги, друзья!

27 июня 1984 года вышел приказ Минвуза РСФСР №410, в котором говорилось: «... организовать в Уфимском нефтяном институте кафедру вычислительной техники и инженерной кибернетики», а 17 июля 1984 года вышел приказ №78-1 по институту, в котором кафедре дано сокращенное название – кафедра кибернетики и присвоена аббревиатура – ВТИК. Кафедра была включена в состав факультета автоматизации производственных процессов. Приказ подписал Ректор УНИ профессор А.И. Сливак.

В 1997 году кафедра стала выпускающей. В 2002 году состоялся первый выпуск инженеров - программистов, в 2010 - первый выпуск бакалавров, а уже в 2012 - первый выпуск магистров направления Информатика и вычислительная техника. Направление

Рис. 1. Главная страница

Данная тема, благодаря более крупным шрифтам и готовым мобильным стилям, придает OpenJournalSystems более современный вид.

Были созданы различные направления, по которым журнал может принимать научные статьи.

- Системы и методы защиты компьютерной информации
- Моделирование информационных систем
- Искусственный интеллект

- Встраиваемые системы
- Системы автоматизированного проектирования
- Геоинформационные системы
- Суперкомпьютеры, параллельные и облачные вычисления
- Системное программирование и техническая кибернетика
- Сети и телекоммуникации
- Информационные технологии в науке и образовании
- Актуальный вопрос науки и технологи

На данный момент журнал уже размещен на домене кафедры вычислительной техники и инженерной кибернетики, где тестируются его модули и разрабатывается окончательный дизайн.

Название Раздела	Аббревиатура	Действие	
Системы и методы защиты компьютерной информации	Защита информации	Редактировать	Удалить ↑ ↓
Моделирование информационных систем	Информ. системы	Редактировать	Удалить ↑ ↓
Искусственный интеллект	Искусственный интеле	Редактировать	Удалить ↑ ↓
Встраиваемые системы	Встраиваемые системы	Редактировать	Удалить ↑ ↓
Системы автоматизированного проектирования	САПР	Редактировать	Удалить ↑ ↓
Геоинформационные системы	Геоинформ. системы	Редактировать	Удалить ↑ ↓
Суперкомпьютеры, параллельные и облачные вычисления	Суперкомпьютеры	Редактировать	Удалить ↑ ↓
Системное программирование и техническая кибернетика	Кибернетика	Редактировать	Удалить ↑ ↓
Сети и телекоммуникации	Сети и телекоммуника	Редактировать	Удалить ↑ ↓
Информационные технологии в науке и образовании	Информ. технологии	Редактировать	Удалить ↑ ↓
Актуальный вопрос науки и технологий	Актуальные вопросы	Редактировать	Удалить ↑ ↓

1 - 11 из 11 результатов

[Создать раздел](#)

Рис. 2. Разделы журнала

Выводы. Разработанный электронный журнал предоставляет возможность ученым, специалистам и аспирантам Уфимского Государственного Нефтяного Технического Университета и кафедры Вычислительной Техники и Инженерной Кибернетики публиковать результаты своих научных исследований по представленным категориям.

Список использованных источников

1. Willinsky J., Stranack K., Smecher A., MacGregor J. // Open Journal Systems: A Complete Guide to Online Publishing, 2010 – 267с.
2. Lambert M. Surhone, Mariam T. Tennoe, Susan F. Henssonow // Open Journal Systems, 2011 - 128

РАЗДЕЛ 2. СВОБОДНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ В БИЗНЕСЕ

Применение систем с открытым исходным кодом для проектирования технологического процесса механической обработки детали

*Алейникова М.А., канд. техн. наук, доцент,
Подколызина Л.В., канд. пед. наук, доцент,
Махмутов Р. Р.,*

ФГБОУ ВО «Санкт-Петербургский политехнический университет Петра Великого»

Аннотация. В статье рассматриваются возможности применения систем с открытым исходным кодом для проектирования технологического процесса механической обработки детали.

Ключевые слова: САПР, открытый исходный код, процесс

Проектирование технологического процесса создания и обработки деталей машин позволяет выявить наиболее рациональный и экономически выгодный метод и способ обработки. Спроектированный процесс механической обработки детали на металлорежущем станке должен соответствовать всем требованиям, предъявляемым к точности и чистоте обрабатываемых поверхностей, правильности контуров и форм.

Сегодня сложную и ответственную работу конструкторов, проектировщиков, дизайнеров и других участников проектирования технологического процесса частично упрощают и делают более эффективной системы автоматизированного проектирования (САПР). САПР реализует информационную технологию выполнения функций проектирования [1], представляет собой организационно-техническую систему, предназначенную для автоматизации процесса проектирования, состоящую из персонала и комплекса технических, программных и других средств автоматизации его деятельности [2]. Первая российская система автоматизированного проектирования была разработана в конце 1980-х годов рабочей группой Челябинского политехнического института, под руководством профессора Кошина А. А..

Для нас наиболее интересным является практика использования компьютерных технологий в проектировании, а именно применение САД системы (англ. computer-aided design), которая не является полным эквивалентом САПР как организационно-технической системы [3]. САД является средством автоматизированного проектирования, предназначенным для решения инженерных задач по построению

двумерных и трехмерных чертежей, создания конструкторской и технологической документации. Компьютерные технологии здесь используются как помощники в создании, модификации, анализе или оптимизации технологической конструкции [4]. Для повышения качества выполнения дизайна, улучшения взаимосвязей посредством ведения документации, а также для создания базы данных технологического процесса для производства используется программное обеспечение CAD. Рассматриваемые системы имеют широкое применение во многих областях производственной деятельности: EDA – системы автоматизированного проектирования электронных приборов, MDA – системы автоматизированного проектирования механических конструкций, где используется векторная графика для изображения объектов традиционной разработки и растровая графика для демонстрации общего вида проектируемых объектов. В результате получаем пакет технических и инженерных чертежей, состав и структуру материалов, размеры, допуски, 2D и 3D фигуры. Это комплексные специализированные системы широкого функционала с серьезным подходом к обеспечению надежности и информационной безопасности.

Наряду с рассмотренными выше системами имеют место и CAD системы с открытым исходным кодом, т.е. предназначенные «Для Всех» - доступное свободное программное обеспечение автоматизированного проектирования процессов и конструкций [5].

Рассмотрим некоторые из таких систем.

QCAD является свободным, открытым исходным кодом для вычерчивания с помощью компьютера конструкций в двумерном измерении (2D). Разработчиком является RibbonSoft GmbH, написана на C++, имеет Qt интерфейс. В системе QCAD можно создавать технические чертежи: планы зданий, интерьеров, механических частей или схем и диаграмм. QCAD работает на Windows, Mac OS X и Linux. Исходный код QCAD выпущен под GPL версии 3 (GPLv3), популярной лицензии Open Source. Пример применения QCAD 3.14 под Windows представлен на рисунке 1.

Система QCAD разработана на принципах модульности, расширяемости и портативности. Отличительной особенностью QCAD является интуитивно понятный пользовательский интерфейс, система проста в использовании, не требует специальной подготовки, и в то же время это мощная 2D CAD система, выполняющая широкий круг задач, распространяется бесплатно. Система QCAD представлена следующими слоями: имеет группировку в виде блоков, включены 35 CAD шрифтов, поддерживаются TrueType шрифты, имеет различные метрические и имперские единицы, DXF и DWG вход и выход, возможность печати в различных масштабах, на нескольких страницах, имеет более 40 строительных инструментов и более 20 их модификаций, возможности

строительства и изменения точек, линий, дуг, окружностей, эллипсов, сплайнов, полилиний, текстов, размеров, штриховки, заливки, растровых изображений, имеет различные инструменты выбора главного объекта, имеет библиотеку измерительных инструментов с более чем 4800 CAD частей, очень полный и очень мощный ECMAScript интерфейс сценариев, поддерживаются форматы DWG, DFX.

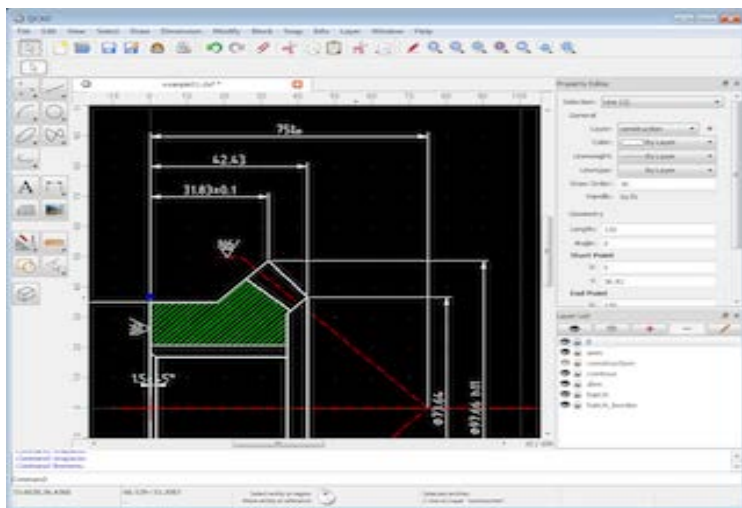


Рис. 1. Применение QCAD 3.14 под Windows

QCAD/CAM - является CAM (Computer-Aided Manufacturing) надстройкой для QCAD, система с открытым исходным кодом 2D CAD. Программное обеспечение QCAD/CAM включает в себя все инструменты и функции, которые QCAD Professional имеет - но кроме того, имеет некоторые специальные инструменты для экспорта чертежей в машиночитываемых форматах файлов, таких как G-Code. Эти файлы обычно используются для управления машинами с компьютерным управлением (ЧПУ), такие как фрезерные станки, гравировальные станки, станок для лазерной резки, плазменной резки или резки водяной струей.

A9Tech - бесплатная CAD-система, которая позволяет создавать чертежи. Программа не обладает таким мощным функционалом как AutoCAD, но инструментов достаточно для воплощения несложных чертежей. Интерфейс представлен на рисунке 2.

Программа имеет стандартный интерфейс: панель инструментов, рабочую область, панель свойств объектов.

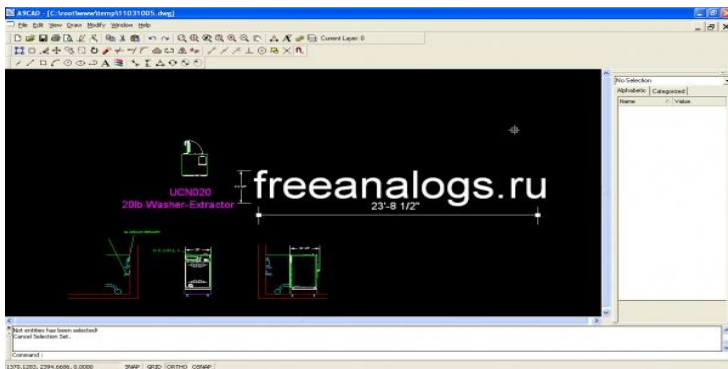


Рис. 2. Окно интерфейса системы А9Тех

А9Тех располагает следующими инструментами для создания чертежей: точки, линии, прямоугольники, дуги, окружности и эллипсы, полигоны, текст, растровые изображения. Для обозначения размеров имеются различные выноски, стиль которых можно настраивать. Над объектами можно совершать стандартные действия: перемещать, изменять размер, вращать и дублировать. Меню со списком инструментов А9Тех представлено на рисунке 3.

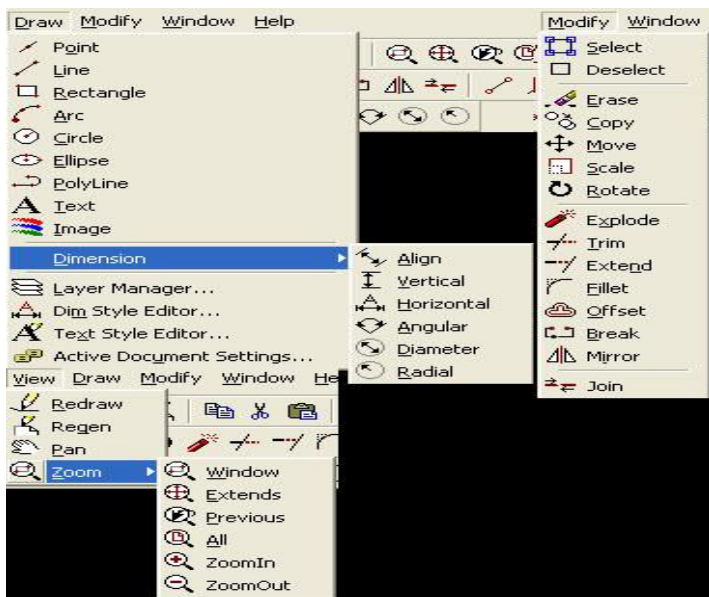
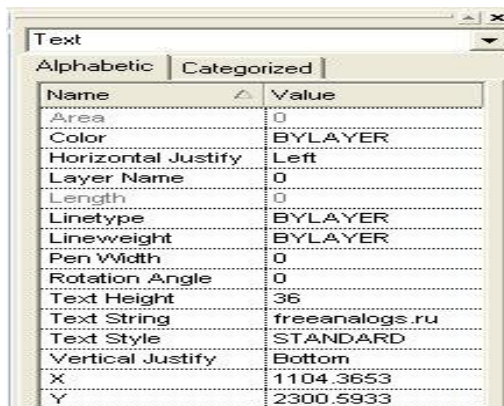


Рис. 3. Список инструментов А9Тех

В качестве примера на рисунке 4 представлен набор свойств объекта Text.



Name	Value
Area	0
Color	BYLAYER
Horizontal Justify	Left
Layer Name	0
Length	0
Linetype	BYLAYER
Lineweight	BYLAYER
Pen Width	0
Rotation Angle	0
Text Height	36
Text String	freeanalogs.ru
Text Style	STANDARD
Vertical Justify	Bottom
X	1104.3653
Y	2300.5933

Рис. 4. Окно выбора объекта Text

Следует сказать, что выбор, выделение объектов и работа с ними не всегда интуитивно понятны пользователю. Каждый объект имеет свой набор свойств, которые можно посмотреть и изменить через панель свойств.

A9Tech позволяет загружать и сохранять файлы DWG и DFX форматов. Результат работы системы можно распечатать на принтере.

Таким образом, можно резюмировать, что система A9Tech - небольшая программа, которая поможет каждому желающему создавать небольшие чертежи.

Использование дополнительных возможностей специализированных программных продуктов САПР - программного обеспечения, предназначенного для создания чертежей, конструкторской и технологической документации, а также 3D-моделей (CAD система для черчения, в САЕ - проектирование твердых тел 3D, САМ - системы инженерных расчетов) позволяет специалистам решать нестандартные задачи любой сложности [6, 7].

Таким образом, спроектированный технологический процесс должен обеспечивать выполнение требований, обуславливающих нормальную работу собранной машины.

Список использованных источников

1. ГОСТ 34.003-90«Информационная технология. Комплекс стандартов на автоматизированные системы. Термины и определения»
2. ГОСТ 23501.101-87«Системы автоматизированного проектирования. Основные положения»

3. Махмутова М.В., Махмутов Р.Р. Разработка проектных решений по внедрению системы управления взаимоотношениями с клиентами VTIGER CRM // В сборнике: Современные инновации в науке и технике Сборник научных трудов 4-ой Международной научно-практической конференции: в 4-х томах. Ответственный редактор Горохов А.А.. 2014. С. 94-98.

4. Махмутова М.В., Подколызина Л.В., Махмутов Р.Р. Применение инновационных образовательных технологий в изучении основ информационной безопасности систем организационного управления / В сборнике: Информационная безопасность и вопросы профилактики киберэкстремизма среди молодежи // Материалы внутривузовской конференции. Под редакцией Г.Н. Чусавитиной, Е.В. Черновой, О.Л. Колобовой. 2015. С. 297-305.

5. Махмутова М.В., Махмутов Г.Р. Модели и платформы реализации массовых открытых онлайн курсов // Современные информационные технологии и ИТ-образование. 2015. Т. 1. № 11. С. 486-496.

6. Махмутова М.В., Махмутов Г.Р. Методика применения методов программной инженерии на этапах разработки информационной системы // Современные информационные технологии и ИТ-образование. 2010. Т. 1. № 6. С. 485-490.

7. Махмутова М.В. Формирование модели образовательной информационной среды подготовки специалиста // Сборник научных трудов Sworld.-2007. Т. 14.-№ 4.-С. 85-90.

Разработка моделей стоимости недвижимости в Нейросимулятор 2.0

Вахрушев В. И.,

Гусева Т. Ф.,

Дьяков В. А.,

ФГБОУ ВПО «Магнитогорский государственный технический университет им. Г.И.Носова»

Аннотация. Статья посвящена разработке нейросетевой модели стоимости недвижимости в г. Магнитогорск в свободно-распространяемой среде Нейросимулятор 2.0. Доказана применимость данного программного инструмента для решения задачи аппроксимации.

Ключевые слова: нейронные сети, стоимость недвижимости, Нейросимулятор 2.0.

Стоимость жилой недвижимости складывается из многих факторов и обычно производится на основе статистических данных. При

этом могут использоваться как классические методы математической статистики, так и современные подходы интеллектуального анализа данных [2, 3, 9].

Среди первых наиболее популярны методы, основанные на анализе статистических характеристик нормального закона распределения, и корреляционно-регрессионный анализ. Несомненным преимуществом указанных методов является их простота, однако они не обладают необходимой гибкостью за пределами ограниченных условий. К ограничениям данных методов относится также и невозможность учета категориальных данных (например, район, этаж, тип домов), существенно влияющих на точность результата оценки [4, 8, 10].

В этом отношении весьма перспективны методы теории искусственного интеллекта и интеллектуального анализа данных, особое место среди которых занимают нейронные сети [6, 11].

Преимущества нейросетевого подхода состоят в следующих особенностях:

- ускорение процесса расчета, которое достигается установлением прямой связи между «входами» и «выходами» (без построения соответствующей математической модели);
- нейросетевой анализ не предполагает никаких ограничений на характер входной и выходной информации;
- самообучаемость нейронных сетей (это особенно важно для слабоформализуемых задач), а также возможность работы с «зашумленными» данными [9].

Дополнительным преимуществом является то, что современные нейропакеты (нейросимуляторы) достаточно просты в применении, а построенные на их основе интеллектуальные системы позволяют решать задачи прогнозирования до некоторой степени надежности и точности. В настоящее время разработано большое количество симуляторов нейронных сетей, выпускаемых рядом фирм и отдельными исследователями, которые позволяют конструировать, обучать и использовать нейронные сети для решения практических задач. Существующие решения можно разделить на три группы: надстройки для программ прикладных вычислений, универсальные и специализированные средства. Для решения поставленной задачи было выбрано программное средство Нейросимулятор 2.0 [1, 12].

Нейросимулятор 2.0. - свободно распространяемый пакет, который является разработкой пермской школы искусственного интеллекта [5]. Нейросимулятор обладает простым, интуитивно-понятным, настраиваемым, удобным интерфейсом и хорошо зарекомендовал себя в образовательном процессе. Каждое действие снабжено пиктограммой и кратким пояснением. В программе также присутствуют средства автоматизированного анализа данных, такие как: определение

значимости параметров и обнаружение выбросов в обучающей выборке. По мнению разработчиков, Нейросимулятор может быть применён не только в образовательных целях, но и в исследовательских и практических задачах [12].

Источником входных и выходных данных являлся сайт «Ситистар» (недвижимость г. Магнитогорска). В качестве параметров были выбраны: количество комнат, тип дома, район, этаж, этажность дома, общая площадь, жилая площадь, площадь кухни, наличие ремонта, мебели, застеклённость балкона и стоимость.

Входные параметры:

X1 – количество комнат;

X2 – тип дома (1 – УП (улучшенная планировка); 2 – БР («брежневка»); 3 – ХР («хрущевка»); 4 – СП (старая планировка); 5 – МС (квартира малосемейного типа); 6 – студия; 7 – новостройка; 8 – нестандартная; 9 – «сталинка»; 10- стандартная; 11 – Свердловский вариант);

X3 – район (1 – Ленинский; 2 – Правобережный; 3 – Орджоникидзевский);

X4 – этаж;

X5 – всего этажей;

X6 – общая площадь (кв. м.);

X7 – жилая площадь (кв. м.);

X8 – площадь кухни (кв. м.);

X9 – балкон (1 – застеклён; 2 – не застеклён);

X10 – наличие мебели (1 – есть; 2 – нет);

X11 – наличие ремонта (1 – есть; 2 – нет).

Выходной параметр: Y1 – стоимость квартиры (тыс. рубл.).

В ходе выполнения работы была сформирована выборка из 200 объявлений, составленная в офисном пакете MSExcel и импортированная в Нейросимулятор.

После определения количества обучающих элементов в выборке мы определили допустимый диапазон нейронов на скрытых слоях персептрона. Для этого мы использовали формулу, являющуюся следствием теорем Арнольда–Колмогорова–Хехт–Нильсена.

$$N = \frac{N_w}{N_x + N_y} \cdot \frac{N_y \cdot Q}{1 + \log_2 Q} \leq N_w \leq N_y \cdot \left(\frac{Q}{N_x} + 1 \right) \cdot (N_x + N_y + 1) + N_y;$$

где N_x – размерность входного сигнала; N_y – размерность выходного сигнала; Q – число элементов обучающей выборки.

Следовательно, число нейронов в скрытых слоях $1.01 \leq N \leq 15.11$ при $N_x=8$, $N_y=1$, $Q=100$.

При определении $N_{\text{опт}}$ методом перебора количества нейронов и обучения персептрона на 2000 эпохах было найдено необходимое количество нейронов в скрытых слоях равное трём.

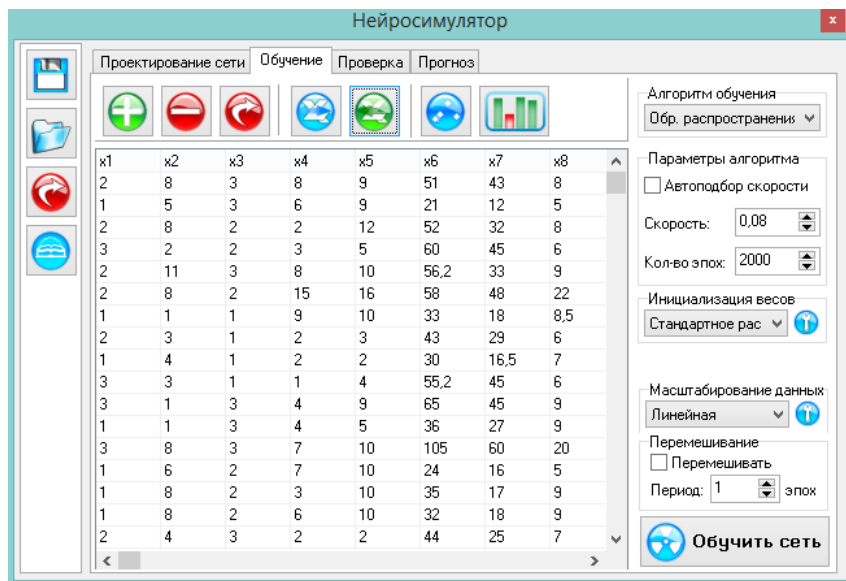


Рис. 1. Фрагмент импортированной выборки.

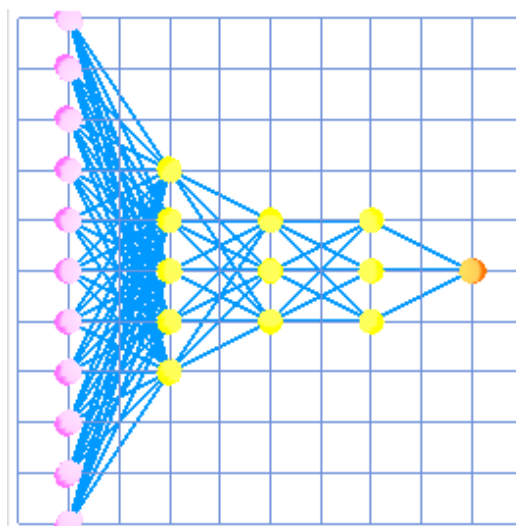


Рис. 2. Графическое представление сети.

Итак, при разработке архитектуры нейросети были установлены следующие параметры:

1. Число слоев нейронов – 5 (3 скрытых).
2. Число нейронов входного слоя – 11.
3. Число нейронов выходного слоя – 1.
4. Число нейронов в первом скрытом слое – 4.
5. Число нейронов во втором скрытом слое – 3.
6. Число нейронов в третьем скрытом слое – 3.
7. Функция активации – сигмоида.
8. Инициализация весов – стандартное распределение.

По результатам обучения нейронной сети была получена среднеквадратичная ошибка 0,3%, которая по нашим меркам является допустимой для проведения прогноза (рисунок 3).



Рис. 3. Распределение Гаусса

Таким образом, квартира со следующими параметрами:

- Однокомнатная
- Брежнёвка
- 3 этаж
- 5-ти этажный дом
- Общей площадью – 30 кв.м.
- Жилая площадь – 17 кв.м.
- Площадь кухни – 6 кв.м.
- С ремонтом
- С мебелью
- С застеклённым балконом

- будет стоить 1060 тыс. рублей, что соответствует действительности.

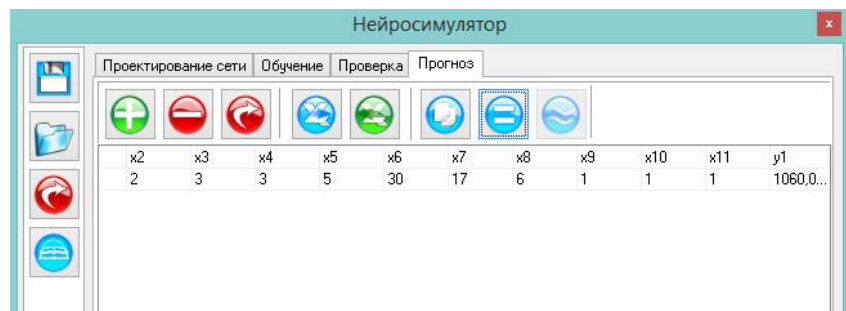


Рис. 4. Прогноз стоимости квартиры.

Для отдельных значений тестовой выборки максимальная ошибка составляла до 14,8%, что может объясняться:

1. Недостаточным объемом исходной информации, например, отсутствие данных о состоянии квартиры.
2. Слабой корреляцией между отдельными параметрами.

Как видно аппарат нейронных сетей, реализованный в Нейросимулятор 2.0., предоставляет удобный инструментарий для решения задач аппроксимации.

Список использованных источников

1. Гаврилова И.В. Методические рекомендации для преподавателей по проведению занятий и оценке знаний студентов по дисциплине "интеллектуальные информационные системы / И.В. Гаврилова. - Магнитогорск, 2011. – 65 с.
2. Курзаева Л.В. Введение в анализ данных с использованием информационных технологий : учеб.-метод. пособие / Л.В. Курзаева, И.Г. Овчинникова. – Магнитогорск : МаГУ, 2012. – 60 с.
3. Курзаева Л.В. Введение в теорию систем и системный анализ : учеб.пособие / Л.В. Курзаева. – Магнитогорск : МаГУ, 2015. – 211 с.
4. Курзаева Л.В. Применение метода попарных сравнений для определения функции принадлежности нечеткой переменной в задачах управления социально-экономическими системами // Научно-практический журнал «Заметки ученого». - 2015 - №5. - С.87-90
5. Курзаева Л.В. Дистанционный курс «Основы математической обработки информации»: электронный учебно-методический комплекс //Хроники объединенного фонда электронных ресурсов «Наука и образование». – 2014. – Т. 1, № 12 (67). – С. 117.

6. Курзаева Л.В. Дистанционный курс «Инструментальные методы поддержки принятия решений»: электронный учебно - методический комплекс//Хроники объединенного фонда электронных ресурсов Наука и образование. 2016. № 1 (80). С. 2.

7. Курзаева Л.В. Нечеткая логика и нейронные сети : учебно-наглядное пособие. – 2015.- 125 с.

8. Курзаева Л.В. Применение метода попарных сравнений для определения функции принадлежности нечеткой переменной в задачах управления социально-экономическими системами / Л.В. Курзаева, Т.Б.Новикова, Ю.С.Лактионова, В.Е. Петеляк, // Научно-практический журнал «Заметки ученого». - 2015 - №5. - С.87-90

9. Масленникова О. Е. Основы искусственного интеллекта: учеб. пособие/О. Е. Масленникова, И. В. Гаврилова. -2-е изд., стер. -М.: ФЛИНТА, 2013. -282 с.

10. Овчинникова И.Г. Математическое обеспечение системы оценки рыночной стоимости недвижимости на основе методов нечеткой логики/ И.Г. Овчинникова, Л.В. Курзаева, В.Е. Петеляк, И.В. Гаврилова // Успехи современной науки и образования. - 2016. - № 3, Том 2. - С. 58-60

11. Попова И.В. Технологии поддержки принятия решений /И.В. Попова, Г.А. Лисьев. – М.:Флинта, 2011. – 133 с.

12. Черепанов Ф.М., Ясницкий Л.Н. Симулятор нейронных сетей «Нейросимулятор 1.0». // Свидетельство об отраслевой регистрации разработки №8756. Зарегистрировано в Отраслевом фонде алгоритмов и программ 12.07.2007

Автоматизация процесса управления бизнес-процессами предприятия

*Кириллов Д.В., Соколова А. А.,
ФГБОУ ВО «Магнитогорский государственный технический
университет им. Г.И. Носова*

Аннотация: В данной статье затронута проблема автоматизации процесса управления бизнес-процессами предприятия. Рассмотрены основные пути решения проблемы. Приведены примеры программных решений, которые позволяют управлять рабочим процессом компании. Описана одна из крупных российских разработок в этой области – система «Битрикс 24».

Ключевые слова: бизнес-процессы, информационная система, автоматизация, веб-приложение, облачный сервис, конфигурация.

В современных условиях, эффективное управление бизнесом представляет собой ценный ресурс организации, наряду с финансовыми,

материальными, человеческими и другими ресурсами. Повышение эффективности управленческой деятельности становится одним из направлений совершенствования деятельности предприятия в целом [1]. В век развитых технологий и активной автоматизации многие предприятия, в том числе и в сфере малого бизнеса, уходят от привычного ручного и речевого управления бизнес-процессами компании, отдавая предпочтение автоматизированному.

Сегодня, можно встретить огромное количество подобных систем и приложений, разработанных как в России, так и за рубежом. Существует небольшое количество комплексных, крупных систем, которые могут обеспечить управление бизнес-процессами компании. Зачастую такие системы представляют ERP-системы, которые внедряют крупные компании. Ярким представителем таких систем за рубежом является ИС Lawson M3 ERP. На российском рынке ярким представителем является система ELMA.

Так как комплексные ERP-системы обладают широким функционалом, не каждая организация готова использовать для себя такое решение. Для разных типов и масштабов предприятий наиболее популярными сегодня стали веб-приложения, которые работают по облачному принципу в режиме Онлайн. Удобство таких приложений заключается в простом и удобном функционале, минимальном месте хранения данных, относительно небольшой стоимости, минимальных системных требованиях.

Единственным явным недостатком веб-приложений для управления бизнес-процессами компании, который может повлиять на привычный рабочий процесс организации, является необходимость в постоянном наличии Интернета. Несмотря на то, что сегодня довольно сложно представить себе организацию, в которой возникают проблемы с использованием интернетом, существует ряд типичных проблем: неустойчивое соединение, перебои и т.п., которые могут повлиять на работоспособность веб-приложений и компании в целом.

Типичные веб-приложения управления бизнес-процессами компании имеют достаточно схожий функционал. Большинство приложений имеют следующие функции [2]:

- Обеспечение удобного взаимодействия сотрудников в единой среде.
- Обеспечение контроля выполнения поставленных задач.
- Формирование отчетных документов по ходу выполнения задач.
- Графическое представление процесса выполнения задач в виде диаграмм и графиков.
- Постановка задач каждому сотруднику.

- Отслеживание времени выполнения задач.
- Ранжирование задач.

В зависимости от особенностей разработки веб-приложения, для повышения конкурентоспособности среди похожих решений, функционал может незначительно отличаться.

Для того, чтобы руководство компании приняло решение в пользу внедрения того или иного веб-приложения, проводится тщательный анализ по различным критериям. Самыми главными критериями всегда будут: цена, наличие необходимого функционала, скорость работы и т.д. В таблице 1 приведена сравнительная характеристика самых популярных на сегодняшний день веб-приложений [2, 3]. Критерии оценивания по таблице:

- «5» - реализовано на высшем уровне.
- «4» - реализовано на достаточном для использования уровне.
- «3» - реализовано на среднем уровне.
- «+» - присутствует в системе.
- «-» - отсутствует в системе.

В последней строчке приводится стоимость для решения, необходимого компании.

Система «Битрикс24» значительно выигрывает среди других систем по функционалу и репутации использования, несмотря на высокую стоимость. В России данная система занимает лидирующие позиции среди SAAS-систем управления бизнес-процессами компании.

Таблица 1

Сравнительная характеристика систем управления бизнес-процессами компании

Критерий	Comindwork	IBN	qTrack	Trello	Мегаплан	Битрикс24
Функционал	4	5	3	3	4	4
Скорость работы	4	4	5	5	4	5
Дизайн	4	3	5	4	4	5
Подключение клиентов в систему	+	+	+	+	+	+
Объединение сотрудников в группы	-	+	+	+	+	+
Система оповещений по e-mail	+	+	+	+	+	+
Разграничение доступа к проектам	+	+	+	-	+	+

Критерий	Comin dwork	IBN	qTrack	Trello	Мегапл ан	Битрикс2 4
Вложенность задач	-	+	-	+	+	+
Возможность указать точное время выполнения	-	+	-	+	-	+
Статусы состояния задач	+	+	+	+	+	+
Шаблоны задач	-	-	-	-	+	+
Оповещение о назначенной задаче	+	+	+	+	+	+
Тайм-трекинг	-	+	-	-	+	+
Загрузка файлов	+	+	+	+	+	+
Привязка файлов к задачам	-	+	+	+	+	+
Блоги сотрудников	+	-	-	-	-	-
Наличие полностью бесплатной версии	-	-	-	-	-	-
Стоимость в месяц	+	+	+	+	-	-
	(\$50)	(\$30)	(\$20)	(\$21)	\$80	\$77

Битрикс 24 - это первый российский SaaS-сервис (облачный сервис), в котором объединены классические рабочие инструменты для организации рабочего процесса компании и социального формата коммуникаций между сотрудниками. Битрикс 24 является системой управления корпоративными порталами, которую реализовала компания «1С-Битрикс» [4].

Система «Битрикс 24» является не только облачным сервисом. Компания 1С: Битрикс предоставляет возможность использования данного сервиса как в облачном варианте, так и в коробочной версии.

1С: Битрикс. Корпоративный портал – это коробочная версия системы «Битрикс 24». Коробочная версия устанавливается на сервер, размещенный в вашей компании заказчика или у хостинг-провайдера. Данную версию можно приобрести как отдельно, так и перейти с облачного сервиса [5].

Оба решения имеют достаточно схожий функционал. Главное отличие заключается в способе и объемах хранения информации. Система «Битрикс 24» находится на облачном сервере компании 1С: Битрикс. Соответственно, все данные также хранятся на этом сервере. Выделенный объем облачного пространства зависит от лицензии, которую приобретает компания. Для коробочной версии системы компания сама определяет, где будет храниться, какой объем пространства будет в пользовании и т.п.

Еще одним важным отличием версий системы «Битрикс 24» является наличие у коробочной версии возможности индивидуально настроить логику работы продукта и дизайн. В облачной версии отсутствует возможность настройки дизайна и логики, возможно лишь настроить бизнес-процессы компании и подключить дополнительные модули. Коротко можно сказать, что коробочная версия имеет немного расширенные возможности по сравнению с облачной, но есть небольшой недостаток – это цены на оба решения.

Зачастую, ключевым показателем для выбора внедрения того или иного решения в компанию является цена. Большинство компаний, прежде всего, смотрят на цену продукта, а затем на функционал и прочие критерии, влияющие на выбор. В случае с системой «Битрикс 24» также необходимо взглянуть на ценовые предложения, перед принятием решения, какую версию лучше приобрести. В таблице 2 приведены стоимости версий системы «Битрикс 24».

Таблица 2
Функциональные и ценовые различия версий системы Битрикс 24

Возможности продукта	Проект	Проект + руб./мес.	Команда руб./мес.	Компания руб./мес.	Корпоративный портал
Стоимость	Бесплатно	990	5 490	10 990	219 500
Пользователи	12	24	неограниченно		25+
Место	5 Гб	24 Гб	100 Гб	неограниченно	
Управление задачами, проектами	+	+	+	+	+
Диаграмма Ганта	+	+	+	+	+
Контроль сроков	+	+	+	+	+
Отчеты	+	+	+	+	+
Автоматизация бизнес-процессов	-	+	+	+	+

Возможности продукта	Проект	Проект + руб./мес.	Команда руб./мес.	Компания руб./мес.	Корпоративный портал
Интеграция с 1С	-	-	+	+	+
Календари (общие и персональные)	+	+	+	+	+
Рабочие отчеты руководителю	-	-	+	+	+
Управление бизнес-процессами (визуальный конструктор)	-	+	+	+	+

По данным таблицы можно увидеть, что у системы «Битрикс 24» существует как бюджетные варианты для малых предприятий, так и дорогостоящие – для крупных компаний.

Для каждой версии, коробочной и облачной, существуют демо-версии, с помощью которых можно протестировать систему и принять решение в пользу той или иной версии. С полной версией можно ознакомиться на официальном сайте компании 1С:Битрикс в разделе Корпоративный портал.

Внешний вид коробочной версии также отличается от облачной. Как уже было сказано ранее, в коробочной версии, за счет предоставления исходного кода и возможности настройки программного продукта под предпочтения и специфику компании, можно полностью переделать систему, пользуясь уже готовыми решениями. В облачной версии же дизайн фиксированный.

Подводя итог, можно сказать, что сегодня сложно представить направление, в котором отсутствует автоматизация, и управление бизнес-процессами компании не является исключением. С каждым годом количество новых программных решений увеличивается, и любая компания может подобрать себе решения, исходя из своих возможностей и предпочтений.

Система «Битрикс 24» является одним из лидирующих решений для автоматизации бизнеса, поддерживая версии для разных предприятий. Если компания нацелена на быстрое внедрение и

небольшую стоимость, то стоит рассмотреть вариант с облачным решением. Обилие разнообразных решений и возможностей для анализа системы позволяет компаниям сделать правильный выбор в пользу системы Битрикс 24.

Список использованных источников

1. Информационный портал для учебных работ «AllBest». Реферат на тему: «Информационные системы управления бизнес-процессами предприятия». [Электронный ресурс]. Точка доступа: http://revolution.allbest.ru/programming/00015198_0.html.

2. Соколова А.А.. Научная работа «Разработка технического задания на создание системы управления процессом разработки программных продуктов «Короб-IT» в компании ООО «ККМ02»//Восьмая международная студенческая электронная научная конференция «Студенческий научный форум 2016». 2016г.

3. Кириллов Д.В. Научная работа «Разработка технического задания на внедрение системы «Битрикс 24»//Восьмая международная студенческая электронная научная конференция «Студенческий научный форум 2016». 2016г.

4. Официальный сайт системы управления проектами «Битрикс24». [Электронный ресурс] - Режим доступа: <https://www.bitrix24.ru/>.

5. Официальный сайт продукции 1С:Битрикс. 1С: Битрикс. Корпоративный портал. [Электронный ресурс]. Точка доступа: <http://www.1c-bitrix.ru/download/intranet.php>.

Диалоговая система "MPRIORITY 1.0"

*Мавлянов Т. Б.,
ФГБОУ ВО «Магнитогорский государственный технический
университет им. Г.И. Носова»*

Аннотация. В статье описан пример решения задачи выбора лучшего CASE-средства с помощью свободной системы поддержки принятия решений MPRIORITY. Приводятся критерии отбора систем, а также отчеты, формируемые системой.

Ключевые слова: система поддержки принятия решений, метод анализа иерархий, MPRIORITY, CASE.

Диалоговая система "MPRIORITY 1.0" (My Priority) предназначена для поддержки принятия решений в различных сферах человеческой

деятельности. "MPRIORITY 1.0" может стать незаменимым помощником для руководителей фирм, подразделений, лабораторий и, всем, кто желает или вынужден по роду своей деятельности принимать обоснованные рациональные решения [5, 7].

Программная система базируется на зарекомендовавшем себя на практике Методе Анализа Иерархий (МАИ). Основное назначение метода - решение слабоструктурированных задач принятия решений [1].

Для того, чтобы понять и научиться работе с данной программой рассмотрим задачу выбора лучшего CASE-пакета на основании определенных критериев. Для анализа были выбраны такие альтернативы, как CA ERwin, MSVisio, RationalRose, AgroUML.

Основными критериями были выбраны следующие:

- стоимость лицензии;
- дизайн;
- универсальность;
- масштабируемость;
- автоматизация операций;
- графические отчеты.

Затем в режиме работы эксперта сравнили данные критерии по шкале, предложенной Т.Саати [2, 3, 4,5]. Пример шкалы сравнения выбранных критериев представлен на рисунке 1.

Качественная шкала

Сравниваем объект А СТОИМОСТЬ

с объектом В ДИЗАЙН

Относительно объекта верхнего уровня

ВЫБОР CASE

A по значимости абсолютно превосходит B
<<промежуточное значение>>

A явно важнее B
<<промежуточное значение>>

A значительно важнее B
<<промежуточное значение>>

A незначительно важнее B
<<промежуточное значение>>

A и B одинаково важны

<<промежуточное значение>>

B незначительно важнее A
<<промежуточное значение>>

B значительно важнее A
<<промежуточное значение>>

B явно важнее A
<<промежуточное значение>>

B по значимости абсолютно превосходит A

Да Отмена

Рис. 1. Шкала сравнения критериев

Результаты сравнения критериев показаны на рисунке 2. Программа автоматически заносит результаты сравнения критериев их в таблицу и производит расчеты. По данным таблицы видно, что наиболее приоритетным критерием является автоматизация операций, далее графические отчеты, затем уникальность, затем стоимость лицензии, затем масштабируемость и на последнем дизайн. Из данных результатов можно сделать вывод, что главными являются функциональные критерии.

Работа эксперта

Производим попарные сравнения относительно объекта

ВЫБОР CASE

	1.	2.	3.	4.	5.	6.	Приоритет
1. СТОИМОСТЬ	1	3	1/3	5	1/7	1/3	0,0903
2. ДИЗАЙН	1/3	1	1/3	1/5	1/7	1/5	0,0336
3. ЧН	3	3	1	1	1/9	1/4	0,091
4. МАСШ	1/5	5	1	1	1/3	1/3	0,0795
5. АО	7	7	9	3	1	5	0,497
6. ГО	3	5	4	3	1/5	1	0,2084

СЗ: 7,1765 Применить

ИС: 0,2353 Закреть

ОС: 0,1897 Отмена

Исследовать

Рис. 2. Результаты сравнения критериев

Следующим шагом было попарное сравнение выбранных альтернатив относительно каждого из критериев (пример по стоимости представлен на рисунки 3, 4).

Работа эксперта

Производим попарные сравнения относительно объекта

СТОИМОСТЬ

	1.	2.	3.	4.	Приоритет
1. CA ERWIN	1	9	1	3	0,4342
2. MS VISIO	1/9	1	1/3	1/3	0,0634
3. RATIONAL	1	3	1	5	0,3748
4. AGROUML	1/3	3	1/5	1	0,1273

СЗ: 4,2604 Применить

ИС: 0,0868 Закреть

ОС: 0,0964 Отмена

Исследовать

Рис. 3. Сравнение относительно критерия «стоимость лицензии»

Результаты исследования представлены на рисунке 4.

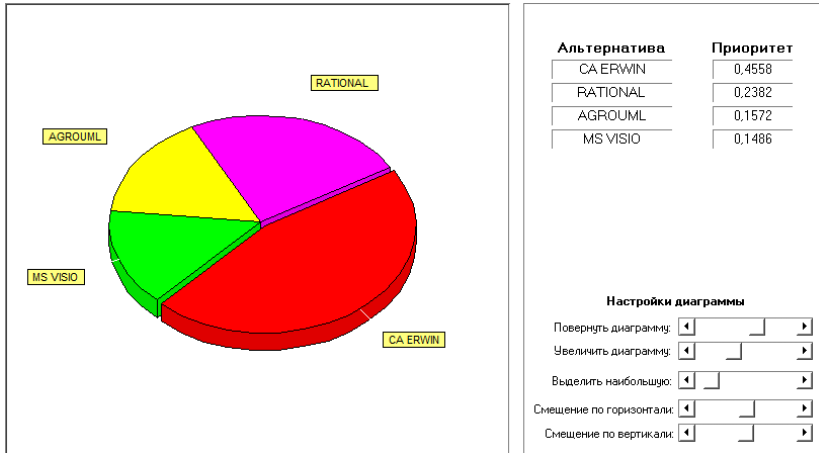


Рис. 4 Результаты исследования

Для того, чтобы полученные результаты были адекватны в ситуациях, в которых принимается решение, необходимо достижение требуемых уровней согласованности данных [6]. Для оценки согласованности вводятся следующие величины: ИС – индекс согласованности; ОС – отношение согласованности.

Для согласованности данных ОС не должно превышать 10%, в исключительных случаях 20%. В данном исследовании в большинстве случаев данные полностью согласованы, то есть ОС=0.

Из данных расчетов следует, что выбор CASE-средства с помощью аналитической системы, основанной на методе анализа иерархий, завершен в пользу программного продукта CA ERwin.

Список используемой литературы

1. Курзаева Л.В. Введение в теорию систем и системный анализ : учеб.пособие / Л.В. Курзаева. – Магнитогорск : МаГУ, 2015. – 211 с
2. Курзаева Л.В. Введение в анализ данных с использованием информационных технологий : учеб.-метод. пособие / Л.В. Курзаева, И.Г. Овчинникова. – Магнитогорск :МаГУ, 2012. – 60 с.
3. Курзаева Л.В. Применение метода попарных сравнений для определения функции принадлежности нечеткой переменной в задачах управления социально-экономическими системами // Научно-практический журнал «Заметки ученого». - 2015 - №5. - С.87-90
4. Курзаева Л.В. Дистанционный курс «Основы математической обработки информации»: электронный учебно-методический комплекс

//Хроники объединенного фонда электронных ресурсов «Наука и образование». – 2014. – Т. 1, № 12 (67). – С. 117.

5. Курзаева Л.В. Дистанционный курс «Инструментальные методы поддержки принятия решений»: электронный учебно-методический комплекс // Хроники объединенного фонда электронных ресурсов Наука и образование. 2016. № 1 (80). С. 2.

6. Курзаева Л.В. К вопросу о совершенствовании методики оценки эффективности решения задач управления качеством образования на основе экспертной информации / Л.В. Курзаева, И.Г, Овчинникова, С.А. Чичиланова // Фундаментальные исследования. 2015. № 6-3. С. 473-478.

7. Попова И.В. Технологии поддержки принятия решений /И.В. Попова, Г.А. Лисьев. – М.:Флинта, 2011. – 133 с.

Обзор пакета Xelopes и его основных возможностей

*Севостьянова А. В., Столяров А И,
ФГБОУ ВО «Магнитогорский технический университет им.
Г.И. Носова»*

Аннотация. В статье описывается структура пакета анализа данных Xelopes, предназначенного для интеллектуального анализа данных. Библиотеки Xelopes открыты, поэтому могут легко встраиваться в любую аналитическую информационную систему. Открытость Xelopes позволяет добавлять к пакету новые алгоритмы анализа данных.

Ключевые слова: Xelopes, анализ данных, datamining.

Современные информационные системы организации немислимы без анализа бизнес-данных, который выступает средством поддержки принятия управленческих решений разного уровня. Качество и широта анализа во многом определяется функциональностью используемых аналитических систем, которые, как правило, проприетарны и дороги. Конечно, существуют недорогие и бесплатные аналоги, но они поддерживают ограниченный набор алгоритмов, т.е. накладывают заметные ограничения на перечень решаемых задач. В то же время существует свободно-распространяемый пакет Xelopes, который поддерживает практически все методы анализа данных, но обладает сложным интерфейсом, который требует некоторого времени на его изучение.

Xelopes – это свободно распространяемая библиотека, разработанная немецкой компанией Prudsys в сотрудничестве со специалистами российской фирмы ZSoft. Архитектура библиотеки соответствует стандарту MDA, что позволило выстроить её в

соответствии с платформонезависимой моделью (Platform-Independent Model — PIM), которая отображается в одну или несколько платформозависимых моделей. Модели PIM создаются с использованием языка UML. Для Xelopes существует три реализации (см. рисунок 1) [1].

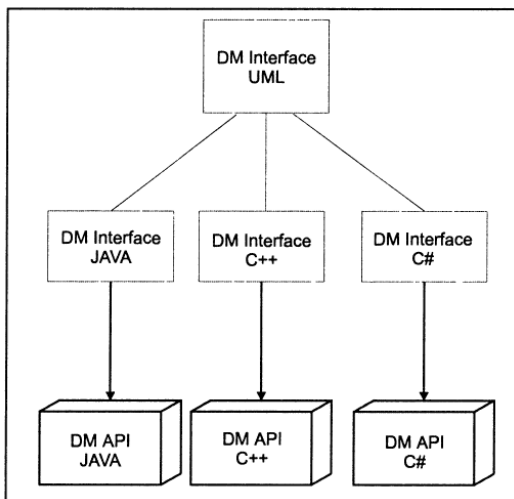


Рис 1. Схема реализации ядра библиотеки Xelopes на трех языках

Благодаря тому, что Xelopes соответствует стандарту CWM, в основу PIM-библиотеки легли диаграммы UML пакета Data Mining этого стандарта. Так, были использованы диаграммы:

- Model - представление Data Mining в целом, классы которого расширяют классы CWM и описывают различные модели;
- Settings - представление настроек процесса Data Mining, классы которого расширяют классы стандарта CWM;
- Attribute - представление описания атрибутов: категориальные, численные, категориально-иерархические и другие, классы которого расширяют классы стандарта CWM.

В библиотеках Xelopes эти диаграммы оснащены новыми классами, методами и интерфейсами, в частности, для формирования формата PMML.

Классы пакета Model описывают различные модели, например, модель сиквенциального анализа, статистическая модель, деревья решений, кластерная модель, модель ассоциативных правил и др.

Классы, реализующие алгоритмы построения различных моделей, включает в себя пакет Algorithms.

Классы унифицированного доступа к разным источникам информации включает в себя пакет DataAccess.

Для того, чтобы библиотека Xelopes могла использоваться во всех процессах Data Mining, были добавлены еще четыре диаграммы UML, которые представляют такие концептуальные области, как: диаграмма DataAccess, которая описывает доступ к данным; диаграмма Algorithms, которая описывает создание моделей Data Mining; диаграмма Automation, которая описывает автонабор параметров алгоритмов Data Mining; диаграмма Transformation, которая определяет описание процесса преобразования данных для моделирования Data Mining. Эти все семь диаграмм являются перечисленным описанием UML ядра Xelopes. Не выпуская из вида тот факт, что с версии 1.1 все классы из пакетов Model, Settings, Attribute, DataAccess, Algorithms, Transformation и Automation основываются на классах стандарта CWM, библиотека совместима с данным стандартом и интегрируется в другие приложения анализа данных благодаря ему.

На основании PIM созданы три основные PSM, получившие реализацию на языках Java, C++ и C#. Также в Xelopes были внедрены адаптеры для популярной библиотеки OLE DB, Weka и Data Mining. Это обуславливает основные достоинства библиотеки Xelopes:

- свободное распространение системы: она находится в открытом доступе и предоставляет возможность разработчикам самим добавлять в нее новые алгоритмы;
- система не зависит от платформы: ее базовое ядро было создано в UML, поэтому реализация может быть получена практически на любом объектно-ориентированном языке;
- независимость от исходных данных: одна из главных идей Xelopes это абстракция – матрица данных. Это сделано потому, что все алгоритмы Data Mining работают в декартовом пространстве переменных, которые характеризуют исследуемые данные;
- поддерживает все основные стандарты в области Data Mining: PMML, CWM, OLE DB for Data Mining, JDM и SQL/MM. Заметим, что в библиотеке имеется реализация адаптеров к популярной библиотеке Weka [1].

Все классы Xelopes расширяются за счёт пакета CWM Core. Это может происходить косвенно - расширением классов из пакетов Data Mining и Transformation - или прямо, когда классы Xelopes расширяют элементы CWM Core.

При решении задач Data Mining с использованием библиотеки Xelopes не важен её вид или используемый метод [2]. Алгоритм решения включает в себя следующие шаги.

1. Загрузка исходных данных.

2. Настройка процесса построения моделей.
3. Инициализация алгоритма.
4. Построение модели.
5. Применение построенной модели для задач с учителем.

Вторая и третья задача зависят от решаемой задачи и применяемого алгоритма.

В заключение можно сказать, что библиотека Xelopes является универсальной основой для стандартного доступа к алгоритмам Data Mining.

Список использованных источников

1. Барсегян А.А. Анализ данных и процессов: учеб.пособие / А. А. Барсегян, М. С. Куприянов, И. И. Холод, М. Д. Тесс, С. И. Елизаров. -СПб.: БХВ-Петербург, 2009. — 512 с.
2. Попова И.В. Технологии поддержки принятия решений / И.В. Попова, Г.А. Лисьев. – М.: Флинта, 2011. – 133 с.

Внедрение системы управления бизнес-процессами для реинжиниринга сельскохозяйственного предприятия

*Ягудина Р. Р.,
ФГБОУ ВО «Магнитогорский государственный технический
университет им. Г.И.Носова»*

Аннотация. В статье оценивается возможность применения систем управления бизнес-процессами на сельскохозяйственных предприятиях. Приводится описание открытой среды RunaWFE, обосновывается целесообразность её применения.

Ключевые слова: система управления бизнес-процессами, сельскохозяйственное предприятие, процессный подход, информационные системы.

Процессный подход к организации бизнес-процессов активно проникает на предприятия любого масштаба любой отрасли народного хозяйства. Специфика деятельности сельскохозяйственных предприятий накладывает особенные требования на системы управления бизнес-процессами. Так, в силу высоких экономических рисков внедряемое решение должно быть недорогим (бесплатным), функциональным, простым в освоении.

Теории процессного подхода можно назвать достаточно зрелыми, так как им посвятили собственные работы как российские, так и иностранные авторы: Абдикеев Н.М., Данько Т.П., Ильдеменов С.В., Киселев А.Д, Калянов Г. Н., Тельнов Ю.Ф., Кловпулос Т., Хаммер М., Чампи Д.

Исследователи полагают, что внедрение процессного подхода повышает эффективность управленческих решений путем формализации повторяющихся последовательностей действий (бизнес-операций) при помощи объединения их в бизнес-процессы. Визуализация деятельности предприятия позволяет быстро перепроектировать бизнес-процессы в ответ на изменение условий деятельности предприятия. Основным условием эффективности внедрения процессного подхода является его автоматизация. Информатизация автоматизация бизнес-процессов позволяет исключить неэффективные процедуры, связанные, как правило, с поиском и передачей информации, а также исключить дублирующие операции и повысить скорость взаимодействия сотрудников [1].

Для автоматизации процессного управления предприятием разработан специальный класс компьютерных систем - системы управления бизнес-процессами и административными регламентами (далее СУБП). Задача этих систем - раздавать задания исполнителям и контролировать их выполнение. В последние годы замечается активное внедрение СУБП, как в государственных организациях, так в бизнесе.

С помощью СУБП можно организовать взаимодействие граждан и различных ведомств посредством электронных каналов связи путем прямого выполнения административных регламентов государственных организаций в компьютерной среде. После работы СУБП в ведомстве появляется аналог производственного конвейера, от которого можно получить наглядное увеличение производительности труда работников, сравнимое с тем, которое было получено от внедрения конвейера на производстве.

Любая система управления бизнес-процессами содержит в себе предназначенный для описания и анализа процессов графический инструментарий, сервер выполнения заданий, на котором выполняются описанные процессы, средства оперативной работы для управления рабочими списками заданий и рабочими приоритетами, а также инструменты мониторинга и управления отображающие стадию и условия выполнения процесса [2, 3].

Кроме увеличения производительности труда СУБП заметно повышают «прозрачность» деятельности предприятия: Все контролируется — сколько прошло времени между обращением гражданина и выполнением задачи, кто выполнил данную задачу, сколько времени выполнял, когда выполнил.

RunaWFE свободно распространяется вместе с исходными кодами на условиях открытой лицензии LGPL. Система бесплатная, без каких-либо ограничений ее можно свободно установить на любое количество компьютеров [4].

В системе RunaWFE для интеграции приложений реализованы специальные сущности - боты и *бот-станции*.

RunaWFE состоит из workflow ядра JBOSS-JBPM и набора дополнительных компонентов, предназначенных для обеспечения удобной работы конечного пользователя. Она поддерживает операционные системы Windows, Linux, Solaris, FreeBSD, веб-сервисы, портлеты, интеграцию с Alfresco и реляционными базами данных, обеспечивает удобный веб-интерфейс пользователя, предоставляет возможность интеграции существующих разнородных приложений предприятия. В её компоненты входят графический редактор бизнес-процессов, боты для выполнения автоматических заданий, гибкая система определения исполнителей на основе ролей, система безопасности, позволяющая интеграцию с LDAP/MS Active Directory [3].

В рамках оценки целесообразности внедрения данного продукта в среде RunaWFE были построены модели бизнес-процессов сельскохозяйственного предприятия и проведена имитация управления ими. В результате исследования было предложено перепроектировать бизнес-процесс и выполнить оценку его производительности. Результаты работы представлены на рисунках 1 и 2.

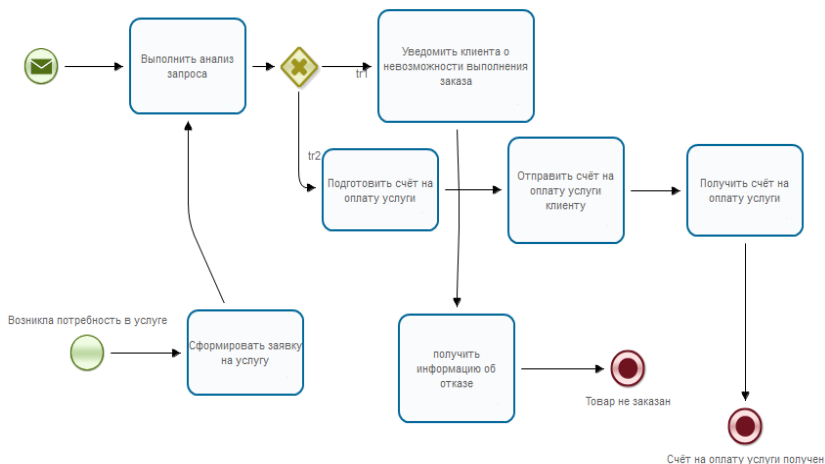


Рис. 1. Модель бизнес-процессов сельскохозяйственного предприятия в нотации BPMN «как есть»

Таким образом, использование системы управления бизнес-процессами RunaWFE позволяет оперативно перестраивать бизнес-процессы организации и административные регламенты сельскохозяйственного предприятия без дополнительных финансовых

рисков. Во многих случаях исполнителей заданий можно даже не информировать об изменении бизнес-процесса, так как это не отразится на характере их работы.

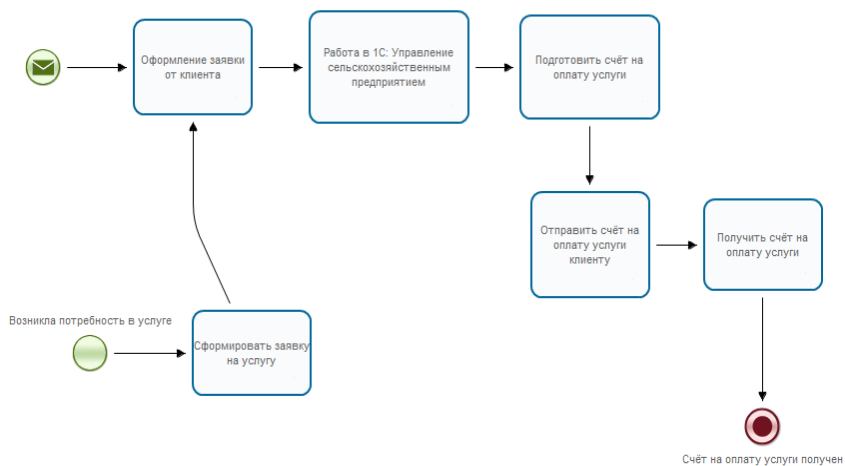


Рис. 2. Модель бизнес-процессов сельскохозяйственного предприятия в нотации BPMN «как должно быть»

RunaWFE дает возможность предприятию эффективнее реагировать на изменение внутренних или внешних экономических условий.

Список использованных источников

1. Михеев А.Г. Практика автоматизации бизнес-процессов на свободном программном обеспечении RunaWFE / А.Г. Михеев . – Режим доступа: <http://www.fa.ru/faculty/mmiar/news/Pages/2015-10-14-praktika-avtomatizacii-biznes-processov-na-svobodnom-programmnom-obespechenii-runawfe.aspx> (дата обращения 15.04.2016);

2. Попова И.В. Свободное программное обеспечение для управления бизнес-процессами / И.В. Попова // Теория и практика применения свободного программного обеспечения Сборник трудов участников Всероссийской молодежной конференции с элементами научной школы. 2011. С. 144-147.

3. Попова И.В. Система управления бизнес-процессами Activity BPM/ И.В. Попова // Теория и практика применения свободного программного обеспечения : Сборник трудов участников Всероссийской молодежной конференции с элементами научной школы. 2011. С. 147-154.

4. Описание RunaWFE [Электронный ресурс]. – Режим доступа: <http://wf.runa.ru>

Построение онтологической модели базы знаний интеллектуальной информационной системы подбора кадров

*Ягудина Р. Р.,
ФГБОУ ВО «Магнитогорский государственный технический
университет им. Г.И. Носова»*

Аннотация. Были проанализированы процессы работы отдела кадров, построена иерархия классов, разработаны атрибуты и свойства экземпляров. Планируется использовать созданную онтологию в качестве системы поддержки принятия решений кадровой службы.

Ключевые слова: поиск, подбор, кадры, онтология, модель, отбор, предприятие.

Работа кадровых служб и руководящего состава любой организации связана с необходимостью поиска и отбора персонала. Отбор кадров является одной из центральных функций управления, поскольку ошибки при отборе новых работников могут приводить к проблемам, серьезно влияющим на работу предприятия. Высокая текучесть кадров, плохой морально-психологический климат, падение трудовой и исполнительской дисциплины (низкое качество работы, прогулы, опоздания на работу и преждевременные уходы с работы, низкая эффективность использования рабочего времени, невыполнение распоряжений руководства и даже саботаж) – всё это последствия неверных управленческих решений в области подбора кадров [5]. В том случае, если речь идет об отборе кандидатов на руководящие должности, ошибки при подборе кадров будут обходиться для организации тем дороже, чем выше уровень должности, на которую отбирается руководитель [2].

Осуществляя подбор сотрудников, необходимо знать и постоянно учитывать особенности современного рынка и ситуацию в экономике, политике, быть в курсе технических и информационных инноваций [2].

Цель работы – изучение и анализ процесса отбора кандидатов на вакантные должности, а также построение онтологической модели базы знаний интеллектуальной информационной системы для подбора кадров, которая будет названа «SistemaPodboraKadrov» на сельскохозяйственном предприятии.

В качестве объекта исследования взято сельскохозяйственное предприятие ГУСП МТС «Зауралье Агро» РБ, занимающееся

деятельностью по производству и реализации кормовых культур, выполнение механизированных работ и предоставление услуг с применением химических средств защиты растений и прочих агрохимических продуктов.

Предметом исследования данной работы является система отбора кандидатов при приеме на работу в предприятие ГУСП МТС«Зауралье Агро» РБ. В ходе работы будет построена с помощью редактора Protege онтологическая модель базы знаний интеллектуальной информационной системы для подбора кадров [6, 7, 8].

В ходе работы первым делом создан класс *SistemaPodboraKadrov*, далее были созданы подклассы класса *SistemaPodboraKadrov*, которые обозначают все этапы в подборе кадров на предприятие. Этими подклассами являются: *PoiskiPodborkandidatov*, *predvaritelnayaBeseda*, *SapolnenieAnkety*, *Sobesedovanie*, *Stagirovka*, *ProverkaRekomendaziy*, *AnalisResultatov*, *SacluchenieDogovora*.

Наглядно класс *SistemaPodboraKadrov* и его подклассы продемонстрированы на рисунке 1.

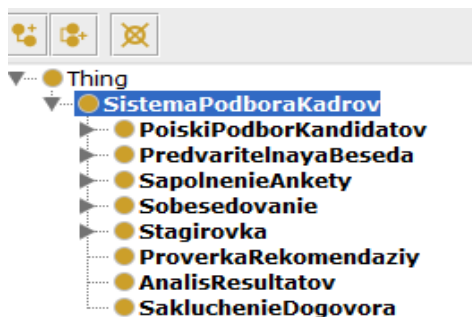


Рис. 1. Иерархия класса онтологии интеллектуальной информационной системы подбора кадров

Добавив подклассы: *PoiskiPodborkandidatov*, *PredvaritelnayaBeseda*, *SapolnenieAnkety*, *Sobesedovanie*, *Stagirovka*, *ProverkaRekomendaziy*, *AnalisResultatov*, *SacluchenieDogovora*, можно указать, что эти подклассы являются непересекающимися, так что любой объект не может быть экземпляром более чем одного из этих трех подклассов. Для указания, что выбранный класс не пересекается с другими классами, необходимо нажать на кнопку «Disjoint with», которая расположена в нижней части панели «Описание класса» рисунка 2.

На этом шаге указали, что эти подклассы являются непересекающимися, так что любой объект не может быть экземпляром более чем одного из этих трех подклассов. Так можно увидеть, что класс

PoiskiPodborkandidatov не является связанным с другими классами это: PredvaritelnayaBeseda, SapolnenieAnkety, Sobesedovanie, Stagirovka, ProverkaRekomendaziy, AnalisResultatov, SacluchenieDogovora.

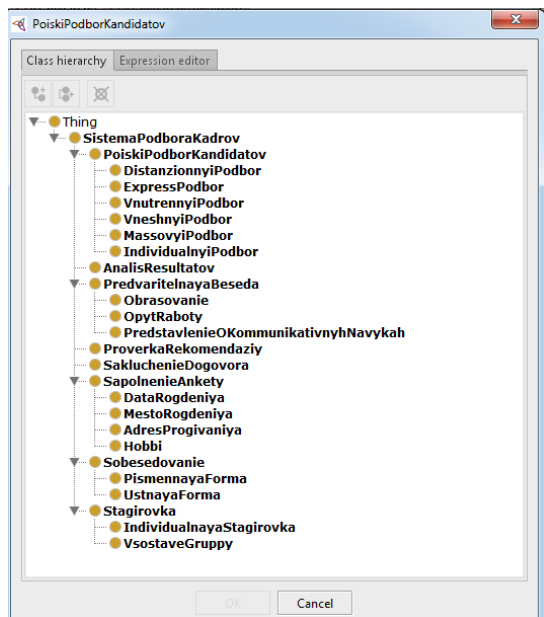


Рис. 2. Описание класса SistemaPodboraKadrov

Для того, чтобы наглядно представить модель онтологии, существует вкладка OntoGraf . Она позволяет графически рассмотреть все связи между классами. Перейдём на вкладку OntoGraf (рисунок 3).

Для того, чтобы подробно рассмотреть следующие классы PoiskiPodborkandidatov, predvaritelnayaBeseda, SapolnenieAnkety, Sobesedovanie, Stagirovka, ProverkaRekomendaziy, AnalisResultatov, SacluchenieDogovora требуется нажать и два раза кликнуть мышью на соответствующем блоке. В итоге будет подробно продемонстрирован требуемый класс.

Основным инструментом отбора кадров выступают анкеты, в которые традиционно включаются вопросы личного характера (дата и место рождения, адрес проживания, образование и т. д.), относящиеся к прежним местам работы, образованию, хобби. Часто присутствуют вопросы, направленные на определение уровня самооценки, отношения к различным явлениям социальной жизни.

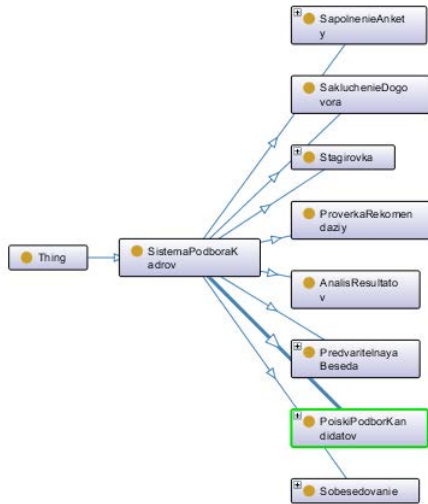


Рисунок 3. Вкладка OntoGraf

На рисунке 4 изображён подробно класс SapolnenieAnkety и входящие в него экземпляры: хобби, дата рождения, место рождения и адрес проживания.

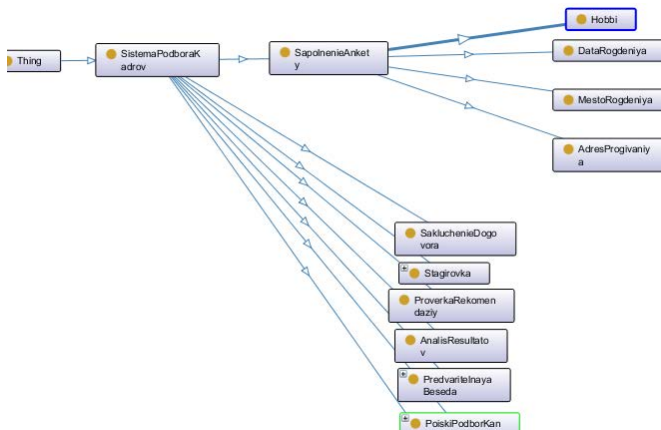


Рис. 4. Графическое изображение класса SapolnenieAnkety

На рисунке 5 изображён подробно класс Stagirovka и входящие в него экземпляры: индивидуальная стажировка и в составе группы. Для каждой вакансии разрабатывается специальная программа тестирования (так называемая «батарея тестов»). Тестирование может быть

индивидуальным или в составе группы, проводится в один день или в течение нескольких.

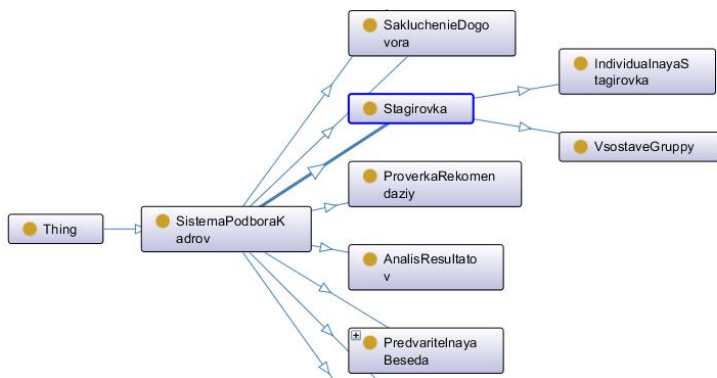


Рис. 5. Графическое изображение класса Stagirovka

На рисунке 6 изображён подробно класс predvaritelnayaBeseda и входящие в него экземпляры: представление о коммуникативных навыках, образование и опыт работы.

На этапе предварительной беседы (по телефону или при встрече) подробнее узнать про образование и, опыт работы кандидата, составить представление о его коммуникативных навыках.

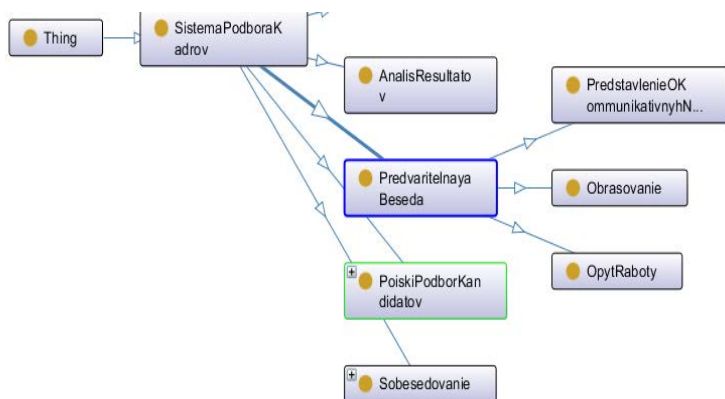


Рис. 6 - Графическое изображение класса predvaritelnayaBeseda

На рисунке 7 изображён подробно класс PoiskiPodborKandidatov и входящие в него экземпляры: массовый подбор, дистанционный подбор,

внешний подбор, внутренний подбор, экспресс-подбор и индивидуальный подбор. На этом этапе происходит определение потребности в персонале и открытие соответствующих вакансий.

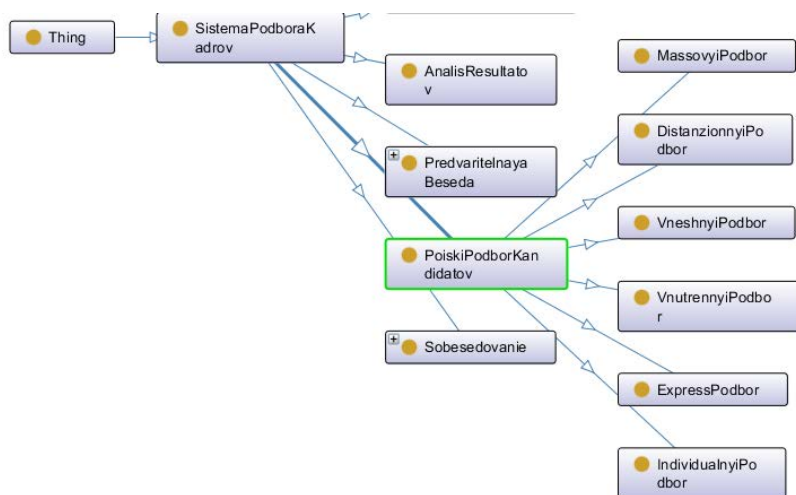


Рис. 7. Графическое изображение класса PoiskiPodborkandidatov

Далее проводится анализ представленных соискателями документов.

Рассмотрим основные разновидности рекрутинга. Бывает точечный, индивидуальный, а бывает массовый подбор, когда нужно найти, подобрать и нанять большое количество сотрудников одновременно. Есть внешний (реализуемый благодаря кадровым агентствам, рекрутинговым компаниям), а есть внутренний подбор (осуществляется внутри фирмы с помощью отдела кадров, службы по работе с персоналом, HR-менеджеров). Встречается экспресс-подбор, который производится в сжатые сроки. Отметим и удаленный или дистанционный отбор (используются современные информационные технологии, различные каналы связи, интернет, при этом личной встречи с кандидатом не происходит).

На рисунке 8 изображён подробно класс Sobesedovanie и входящие в него экземпляры: письменная форма и устная форма. Интервьюирование кандидата, в той или иной мере структурированное и формализованное, может проводиться в устной или письменной форме.

На рисунке 9 представлено изображение онтологической модели базы знаний интеллектуальной информационной системы для подбора кадров.

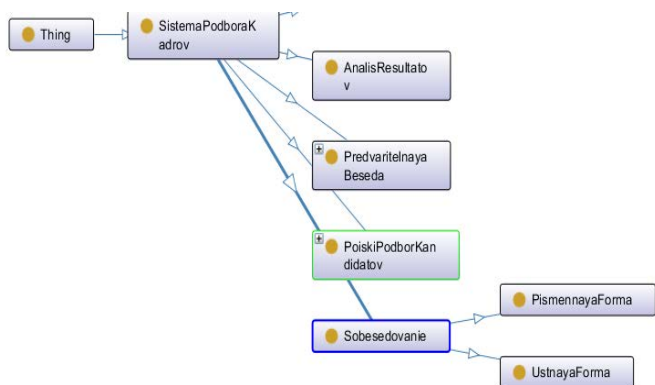


Рис. 8. Графическое изображение класса Sobesedovanie

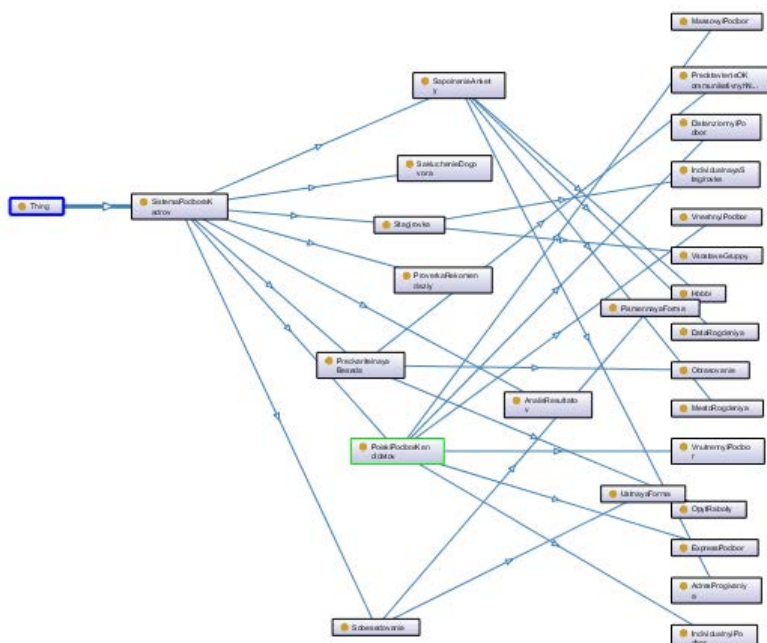


Рис. 9. Графическое изображение модели SistemaPodboraKadrov

В ходе работы были созданы и описаны классы онтологической модели базы знаний интеллектуальной информационной системы для

подбора кадров: oiskiPodborkandidatov, PredvaritelnayaBeseda, SapolnenieAnkety, Sobesedovanie, Stagirovka, ProverkaRekomendaziy, AnalisResultatov, SacluchenieDogovora. Далее были выделены соответствующие экземпляры каждого класса.

Список использованных источников

1. Повышение эффективности подбора персонала за счет автоматизации процесса [Электронный ресурс] – Режим доступа: <http://efsol.ru/articles/recruiting-automation.html> (дата обращения 13.01.2016 год)

2. Подбор персонала [Электронный ресурс] – Режим доступа: <http://www.grandars.ru/college/biznes/podbor-personala.html> (дата обращения 13.01.2016 год)

3. Управление персоналом [Электронный ресурс] – Режим доступа: <http://psyfactor.org/personal/personal15-02.htm> (дата обращения 13.01.2016 год)

4. Оценка эффективности подбора персонала в компании [Электронный ресурс] – Режим доступа: <http://www.ht.ru/cms/component/content/article/1-aricles/702-2009-10-08-13-18-28> (дата обращения 13.01.2016 год)

5. Методы подбора и поиска персонала [Электронный ресурс] – Режим доступа: <http://constructorus.ru/karega/podbor-i-poisk-personala.html> (дата обращения 13.01.2016 год)

6. Попова И.В. Технологии поддержки принятия решений / И.В. Попова, Г.А. Лисьев. – М.: Флинта, 2011. – 133 с.

7. Гаврилова И.В. Свободное программное обеспечение для управления бизнес-процессами / И.В. Гаврилова // Теория и практика применения свободного программного обеспечения Сборник трудов участников Всероссийской молодежной конференции с элементами научной школы. 2011. С. 144-147.

8. Попова И.В. Применение технологии открытых систем для построения эффективных автоматизированных информационных систем/ И.В. Попова // Управление информационными ресурсами образовательных, научных и производственных организаций : материалы Всероссийской научной школы для молодёжи / под общ.ред. А.Я. Олейникова, Г.Н. Чусавитиной, Л.З. Давлеткиреевой. – Магнитогорск : МаГУ, 2009. – С.137-138.

РАЗДЕЛ 3. СВОБОДНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ В ПРОГРАММНОЙ ИНЖЕНЕРИИ

Web-интерфейс вывода статистики системы учета студентов

*Бекоев Б.Г., Шебанин А.С., Сергеев К.А.,
НИУ Высшая школа экономики*

Аннотация. В данной статье рассмотрены основные этапы разработки WEB-интерфейса вывода статистики учета студентов. Приводится описание приложения, которое позволит преподавателю отслеживать посещаемость проводимых занятий. Приводится описание web-интерфейса, который взаимодействует с файловой базой данных и позволяет преподавателю получить информативную статистику по каждой лекции, определенной дате или группе.

Ключевые слова: система учета посещаемости студентов, разработка, web-интерфейс.

Введение. В университете преподаватель ведет занятия в среднем минимум в 3 группах. Бумажный учет постепенно вытесняется использованием тех или иных информационных средств. В большинстве случаев преподаватель использует распечатанные таблицы с именами студентов, после чего ему нужно произвести учет присутствующих на занятии студентов, затем необходимо все эти данные занести обратно в электронную таблицу. Как видно, данный процесс довольно затратный по времени.

Именно по этой причине есть необходимость в автоматизировании процесса обработки учетной информации. Уменьшение времени учета и контроля посещаемости студентов будет способствовать увеличению производительности и снижению трудоемкости решаемых задач.

Также необходимо обеспечить доступность получения данных об учете вне зависимости от типа используемого устройства, как мобильного, так и персонального компьютера.

Задача данной системы – свести время, которое тратится на учет, к минимуму, а также обеспечить возможность получения детальной статистики по каждой группе, занятию, студенту и упростить последующее формирование отчетности.

Обоснование необходимости разработки собственного решения

Основной необходимостью разработки собственной системы учета контроля студентов является отсутствие гибкости во всех существующих продуктах. Невозможность вносить небольшие изменения в исходный код, дорабатывать и подстраивать его под свои нужды.

Как правило, большинство из предложенных на рынке систем были разработаны под конкретную задачу, но, если у пользователей или администраторов системы возникает потребность в новом функционале, или отключении неиспользуемых функций – процесс обновления или изменения продукта займет большую часть времени и, соответственно повлечет за собой финансовые затраты. При возникновении критических проблем, все упирается в организацию-разработчика системы и ожидание его действий по решению проблемы.

Нельзя сказать, что всех проблем можно избежать, разработав свою систему и наняв своего разработчика и администратора, но эти действия сильно снизят риски возникновения каких-либо проблемных ситуаций и время на их решение.

Структура проекта

1. Распределение по занятиям, с возможностью получения статистики по каждому занятию.
2. Распределение по группам, с возможностью получения статистики о каждом студенте в группе.
3. Возможность получения информации о занятии по определенной дате.
4. Возможность вывода детальной статистики о присутствующих.
5. Детальная информация о каждом студенте: время прихода и ухода, проведенное на паре время, номер группы, оценка.
6. Возможность просмотреть преподавателем комментарии студентов о занятии.
7. Полная статистика по всем проведенным занятиям.

Выбор средств реализации

Для реализации поставленной задачи нам нужна система клиент-сервер с централизованным способом хранения данных об успеваемости и возможностью ограничения доступа для разных видов пользователей системы. Для данного проекта было принято использовать платформу Web, основной код backend программы написан на PHP, в качестве веб-сервера будет выступать Apache в связке с Nginx и файловая база данных.

Данный подход имеет следующие преимущества:

- данные хранятся в одном месте и всегда есть возможность получить их на любом устройстве и вне зависимости от используемой ОС;
- вся логика работы хранится на сервере, что защищает персональные данные учащихся от доступа третьих лиц;
- возможность быстрого запуска, развертывания и переноса системы с одного сервера на другой;

- отсутствует необходимость в написании своего клиента для просмотра данных. Для получения необходимой информации достаточно наличие браузера и интернет соединения;

Model-View-Controller. Для разделения логической структуры и реализации компонентов программы будет использован шаблон проектирования (паттерн) model-view-controller. Схема его использования заключается в том, что модель приложения, интерфейс пользователя и взаимодействие с пользователем поделены на три независимых компонента:

- Model (Модель). Модель отдает данные (обычно для View), а также отвечает на запросы (в основном от контроллера), изменяя свое состояние;

- View(Представление). Отвечает за вывод информации.

- Controller (Поведение). Интерпретирует данные, которые ввел пользователь и сообщает модели и представлению о необходимости нужной реакции.

Использование в веб-приложениях

Хотя первоначально он был разработан для настольных компьютеров, паттерн модель-представление-контроллер был широко принят в качестве архитектуры World Wide Web приложений в основных языках программирования. Несколько коммерческих и некоммерческих фреймворков веб-приложения были созданы, используя данный шаблон.

AngularJS [3] - это очень мощная JavaScript [2] библиотека. Она используется в одностраничных приложениях (SPA). Она расширяет HTML DOM с дополнительными атрибутами. AngularJS является проектом с открытым исходным кодом, полностью бесплатен, и используется тысячами разработчиков во всем мире и распространяется под лицензией Apache License версии 2.0.

AngularJS использует HTML для определения пользовательского интерфейса приложения. HTML является декларативным языком, который является более интуитивным и менее запутанным, чем определение интерфейса процедурно в JavaScript. Кроме того, над разработкой пользовательского интерфейса могут одновременно работать много frontend-разработчиков.

Структура используемой файловой базы данных. Нельзя сказать, что использование базы данных реляционного типа во много уступает выбранному решению.

Однако, используя данный тип БД, мы становимся независимыми от сервера БД. Что значительно повысит отказоустойчивость разрабатываемой нами системы. Плюс ко всему, скорость работы с файловой системой значительно быстрее, нежели с СУБД.

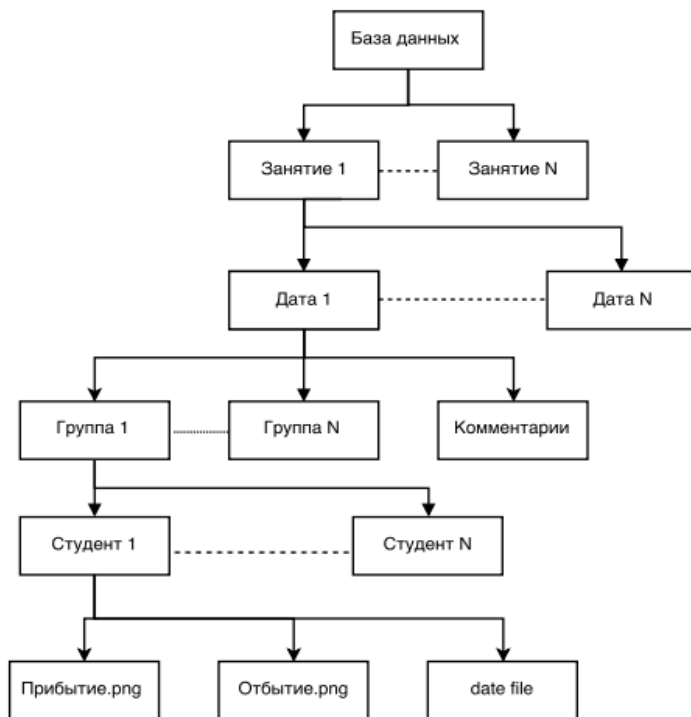


Рис. 1. Структура файловой БД

Описание разработанной системы

Для того, чтобы отметить на занятии, обучающемуся необходимо зайти на веб-форму (рисунок 2) с помощью мобильного телефона или ПК, выбрать дисциплину, группу, ФИО и по желанию оставить комментарий. После этого необходимо сфотографировать себя, что будет доказательством факта присутствия конкретного студента на занятии. Данные о группе и ФИО берутся из БД, что уменьшает количество времени на заполнение личных данных обучающегося.

Интерфейс преподавателя. На первой странице происходит вывод общего количества занятий с датами их проведения и общим количеством пар за эту дату. (рисунки 3,4)

На следующей странице предлагается выбор группы (рисунок 5) для отображения детальной статистики. Также на этой странице отображаются комментарии студентов о прошедшем занятии.

Учет посещаемости

Дисциплина

Программирование

Группа

C-31

ФИО

Шебанин Александр Сергеевич

Комментарий

Не очень ясно объяснили такой принцип ООП, как "наследование"

Фото



Сфотографироваться

Отправить данные

Рис. 2. Веб форма для учета посещаемости

Система учета студентов



Занятия

Название занятия	Дата проведения	Количество пар	Действие
Занятие 2	18-04-2014	1	Подробнее
Занятие 3	16-05-2014	4	Подробнее

Рис. 3. Веб интерфейс для преподавателя

Занятие 3

Группа	Посетили (чел.)	Средний балл	Действие
АП-21	17	5	Подробнее
ИКТ-21	3	4.6	Подробнее
С-21	14	5	Подробнее

Комментарии к занятию

Комментарии отсутствуют

Рис. 4. Веб интерфейс для преподавателя

Рисунок 5 показывает статистику по занятию в определенной группе, при наведении на конкретного обучающегося открывается окно с его фотографиями.

Занятие 3

ФИО	Пробыл на паре	Прибытие	Убытие
Бажинский Алексей Геннадьевич	1ч. 1мин.	13:46:32	14:47:40
Балакичев Александр Борисович	1ч. 0мин.	13:49:48	14:50:38
Батов Андрей Сергеевич	0ч. 54мин.	13:51:08	14:46:04
Бехрев Богдан Сергеевич	1ч. 1мин.	13:45:17	14:47:04
Горюхов Егор Сергеевич	1ч. 4мин.	13:44:38	14:49:33
Дорожков Александр Александрович	1ч. 2мин.	13:50:24	14:53:17
Иванов Александр Иванович	0ч. 59мин.	13:49:07	14:48:13
Корсаков Александр Владимирович	0ч. 59мин.	13:52:20	14:52:02
Корсаков Кристина Валерьевна	1ч. 1мин.	13:47:23	14:48:51
Корсаков Сергей Сергеевич	1ч. 2мин.	13:48:01	14:50:02
Мондров Александр Сергеевич	1ч. 2мин.	13:43:34	14:46:33
Морозов Александр Александрович	1ч. 6мин.	13:45:55	14:52:47

Рис. 5. Веб интерфейс для преподавателя

Заключение. Разработанное программное обеспечение поможет преподавателю не отвлекаться на операции учета посещаемости и может быть использовано для уменьшения трудозатрат преподавателей, за счет уменьшения кол-ва рутинной работы: 1) печать таблиц посещаемости, 2) заполнение таблиц посещаемости, 3) перенос данных о посещаемости в электронный вид. Кроме этого, с его помощью можно вести детальную статистику по каждой группе, студенту и занятию.

Список использованных источников

1. Робин Никсон Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. – Пер. с англ. – СПб: Питер, 2015. – 688 с.
2. Флэнаган Д. JavaScript. Подробное руководство. – Пер. с англ. – СПб: СимволПлюс, 2008. – 992 с., ил.
3. Raoni Boaventura A Step-by-Step Guide to Your First AngularJS App [Электронный ресурс]. – Режим доступа: <http://www.toptal.com/angular-js/a-step-by-step-guide-to-your-first-angularjs-app>

Анализ алгоритма принятия решения об автоматизации тестирования программного продукта с применением свободного программного обеспечения Selenium

Галимова Е. Ю.,

Высшая школа печати и медиатехнологий Санкт-Петербургского государственного университета промышленных технологий и дизайна

Аннотация. В статье рассматривается методика оценки потребности автоматизации тестирования программного продукта в процессе его разработки. Приводятся требования к подобной системе. Описываются особенности системы Selenium и рассматриваются её функциональные возможности. Обосновывается целесообразность применения Selenium для автоматизации тестирования программных продуктов.

Ключевые слова: Selenium, тестирование, автоматизация.

На вопрос о применении автоматизации в процессе тестирования программного продукта (ПП) можно ответить только после проведения тщательного анализа многих характеристик ПП. Обычно окончательное решение принимается на этапе передачи разработчиками ПП в тестирование.

Любой ПП в течение своего жизненного цикла проходит ряд этапов. Основные этапы создания ПП [1] – разработка и утверждение

технического задания, затем эскизного проекта, рабочего проекта и, наконец, внедрение. Этап создания технического задания подразумевает постановку задачи, сбор исходных данных, выбор и аргументацию критериев эффективности и качества разрабатываемого программного продукта, доказательство необходимости проведения научно-исследовательских работ. Определяется структура входных и выходных данных, осуществится выбор методов решения задач, обосновывается необходимость ранее разработанных программ, определяются требования к техническим средствам, доказываемая принципиальная возможность решения поставленной задачи. На этапе разработки и утверждения технического задания необходимо так же определить требования к программе, дать технико-экономическое обоснование разработки, определить этапы и сроки разработки программы и документации, выбрать язык программирования, согласовать и утвердить техническое задание. Этап создания эскизного проекта состоит из проработки структуры входных и выходных данных, детализации методов решения задачи, разработки общего описания алгоритма решения задачи, разработки технико-экономического обоснования и пояснительной записки, согласования и утверждения эскизного проекта. Этап создания технического проекта охватывает детализацию структуры входных и выходных данных, разработку алгоритма решения задачи, выбор вида представления входных и выходных данных, определение семантики и синтаксиса языка, разработка структуры программы, точное определение конфигурации технических средств. Так же на этом этапе выполняется выработка плана мероприятий по разработке и внедрению программ, разработка пояснительной записки, согласование и утверждение технического проекта. На этапе создания рабочего проекта программируется и отлаживается программа, разрабатывается программная документация, разрабатывается, согласуется и утверждается порядок и методика испытаний, проводятся предварительные приемосдаточные и другие виды испытаний, программа и документация корректируются по итогам испытаний. Этап внедрения подразумевает передачу программы и программной документации, оформления акта о передаче программы, передачу программы в фонд алгоритмов и программ.

Для улучшения взаимодействия групп программистов и тестировщиков на этапе передачи ПП в тестирование предлагается ряд вопросов (критериев), ответы на которые должны поступать на тестирование вместе с передаваемым ПП. Эти ответы для ухода от большой работы в группе программистов должны быть краткими (в самом простом виде – «да/нет») и даваться ведущими специалистами этой группы, предпочтительно – архитектором ПП (системным аналитиком). Они, путём диалога с программистами, могут уточняться

после процедуры просмотра списка спецификаций ведущими тестировщиками. На основе полученных ответов в группе тестирования принимается решение о ручном, автоматизированном или смешанном виде тестирования ПП или его модулей (частей).

Список вопросов, положительный ответ на которые склоняет в сторону автоматизации тестирования [2]:

В ПП есть функционал для выполнения повторяющихся действий?

Часто будут выходить новые версии ПП?

Имеется большое количество форм с полями для ввода данных?

Предъявляются высокие требования к производительности?

Будут при работе с ПП производиться переходы с одной конфигурации на другую?

Предполагается эксплуатация ПП на максимальной нагрузке?

В ПП много web-ссылок?

Группа вопросов в пользу ручного тестирования:

Функционал предполагает выполнение ручных действий во время работы ПП?

Планируется проверка эргономичности ПП?

Планируется регулярно проверять корректность установки, обновления и удаления ПП?

Будет проверяться удобочитаемость формата выходных данных?

При разработке использовалось много сторонних управляющих элементов?

Необходимо оценивать способность восстановления системы после сбоя?

В ПП много графических объектов?

В ПП много функционала, который предполагает печать документов на принтере?

Тестирование должно пройти в сжатые сроки?

Группа вопросов в пользу смешанного тестирования:

Планируется проводить функциональное тестирование?

Предполагается создавать наборы входных тестовых данных заново перед каждой итерацией тестирования?

Будет проводиться тестирование на некорректных входных данных?

При разработке ПП использовались сложные логические структуры?

Задача выбора способа тестирования решается как формальная многокритериальная задача [3]. Ответы «да» получают бинарное значение единица, ответы «нет» - ноль. Получается матрица размерности $2 \times N$, где N – количество вопросов. Далее тестировщиком дается вес каждому вопросу. Диапазон весов от 0,25 до 1. Образуется линейная свертка, которая затем делится на количество вопросов. По результату,

лежащему в границах от нуля до единицы включительно, даются рекомендации о выборе ручного, автоматизированного или смешанного подхода к тестированию данного ПП.

В случае принятия решения об автоматизированном или смешанном тестировании ПП (его отдельных модулей) должен быть выбран инструмент автоматизации. Есть инструменты, относящиеся к свободному программному обеспечению. Свободным считается то ПО, которое можно беспрепятственно запускать, копировать, распространять, изучать, изменять, улучшать. Другими словами, программу можно свободно скачать из Интернета и абсолютно легально установить на свой компьютер.

У свободного программного обеспечения есть ряд преимуществ. В первую очередь, финансовый аспект – за него не нужно платить. С другой стороны, переносимость свободного ПО привела к тому, что множество программистов по всему миру работают над его улучшением, устранением ошибок, добавляют новый функционал. Обычно существует дополнительная платная служба поддержки, доступная даже из удаленных точек мира. Свободное программное обеспечение легче адаптировать под нужды конечного пользователя, так как из большого количества доступных приложений легко конструируются новые программы.

В наши дни стали очень часто выходить новые версии ПП, с каждой версией возрастают требования к мощности компьютеров. Зачастую свободное программное обеспечение предъявляет меньшие требования к компьютерным ресурсам, чем его коммерческие альтернативы.

На сегодняшний день существует широкий выбор качественного свободного программного обеспечения, которым пользуются многие крупные международные компании, государственные учреждения, правительственные организации (Google, IBM, BMW, Volvo, Yandex). Более 80% информационных ресурсов в сети интернет управляется свободным программным обеспечением [4].

Большое распространение в наши дни среди профессионалов в сфере информационных технологий получил инструмент Selenium, предназначенный для автоматизации тестирования. Его важными преимуществами стало наличие драйверов под все распространенные платформы, включая мобильные, а также развитый и удобно реализованный функционал. На официальном сайте Selenium доступен полный список существующих команд, что значительно упрощает работу тестировщика.

Разработку тестов на Selenium можно производить в режиме `testrunner` или в режиме `driven` [5]. Первый отличается простотой создания тестов, без использования языков программирования.

Выполнение тестов происходит непосредственно в веб-браузере, что позволяет имитировать действия пользователя. Данный режим рекомендуется для начинающих тестировщиков-автоматизаторов. Тесты, созданные в режиме driven, могут запускаться вне веб-браузеров. С их помощью можно реализовывать более обширный набор функций. Для их создания требуется знание языков программирования.

Selenium состоит из нескольких частей [6]:

- Selenium IDE (Integrated Development Environment) предназначен для написания небольших тестовых сценариев. Позволяет автоматизировать рутинные, повторяющиеся действия в процессе ручного тестирования. Это интегрированная среда разработки, дополнение к веб-браузеру Firefox. Есть удобное контекстное меню. Сначала выбирается элемент, отображаемый на странице веб-браузера, затем – команда из контекстного меню.

- Selenium Web Driver предназначен для написания больших тестовых наборов. Он совместим с любым браузером. WebDriver поддерживает сложную логику разрабатываемых тестов. Он позволяет управлять тестами непосредственно из тестируемого ПП. Этот инструмент имеет удобный программный интерфейс, он легкий и доступный в изучении.

- SeleniumServer позволяет запускать тесты удаленно, он предназначен для разработки большого количества тестовых наборов для одного ПП. Несколько серверов образуют распределенную сеть, которая легко поддается масштабированию. Ее называют Selenium Grid. Сеть упрощает тестирование на различных многочисленных конфигурациях операционных систем, то есть проведение кроссплатформенного тестирования.

Различные приемы проектирования тестов могут использоваться в Selenium. Во-первых, тесты для проверки статичного контента, то есть не изменяющихся элементов в интерфейсе. Можно проверить, верные ли заголовки у страниц, есть ли нужные изображения, в правильных ли местах они расположены. Автоматизировать данный вид тестирования имеет смысл, если содержимое страниц будет со временем меняться.

Selenium поддерживает тестирование ссылок. Производится переход по каждой ссылке и проверка, что мы оказались на нужной странице.

Есть возможность проводить тестирование функциональности. Этот вид тестирования обычно самый трудоемкий, требуется проверить полноту реализованности заявленного функционала. На вход подаются данные, обычно путем заполнения специально предназначенных в приложении полей, на выходе ожидается результат, заявленный в спецификации.

С помощью Selenium легко осуществить тестирование динамических объектов. К ним относятся те, чей идентификатор генерируется при каждом открытии страницы. Для поддержки динамических элементов пользовательского интерфейса разработана специальная технология AJAX, для которой можно разработать тесты.

Selenium поддерживает как десктопные, так и мобильные браузеры, позволяет разрабатывать тестовые сценарии на большинстве популярных языков программирования, он поддерживается производителями всех известных браузеров, позволяет выполнять сценарии в облаках. Будущее автоматизации тестирования ПП тесно связано с развитием свободного программного обеспечения, в том числе Selenium.

Список использованных источников

1. ГОСТ 19.102-77 Единая система программной документации (ЕСПД). Стадии разработки

2. Галимова Е. Ю. Признаки выбора вида тестирования программного обеспечения// Вестник молодых ученых Санкт-Петербургского Государственного университета технологии и дизайна. - 2014. - №1

3. Галимова Е. Ю. Применение алгоритма многокритериальной оптимизации при выборе между ручным и автоматизированным тестированием// 63 Международная молодежная научно-техническая конференция “Молодежь. Наука. Инновации”.- Владивосток, 2015.

4. Свободное программное обеспечение. – Режим доступа: <http://vsfte.biz/184-svobodnoe-programmnoe-obespechenie.html>

5. Автоматические приемочные тесты с Selenium. – Режим доступа: <http://www.ibm.com/developerworks/ru/library/wa-selenium-ajax/index.html>

6. Selenium : сайт. [Электронный ресурс]. - Режим доступа: <http://selenium2.ru/>

Особенности внедрения внутренних информационных разработок в действующую проприетарную информационную систему университета

*Журавлев А. Е., к.т.н., доцент,
ФГБОУ ВО "Государственный университет морского и речного
флота им. адмирала С.О Макарова"*

Аннотация. Представлено описание реализации модуля, интегрирующего существующие автоматизированные информационные

системы университета, и методики его внедрения. Обоснована эффективность разработанного решения.

Ключевые слова: внедрение, интеграция, информационная система, свободное программное обеспечение, университет.

Актуальные методы подготовки специалистов-профессионалов все чаще обращаются к системам электронного (в том числе дистанционного) обучения. С ростом потребности все разнообразнее становится рынок предлагаемых обучающих информационных систем, что, в конечном счете, приводит к высокой фрагментации и децентрализации отдельных компонентов, причем основная масса предложений рынка является проприетарной и найти что-либо, одновременно удовлетворяющее всем основным требованиям и открытое практически невозможно. Так, на базе условно-общего информационного поля университета, в действительности могут функционировать совершенно автономные системы самых различных, иногда антагонистичных архитектур и парадигм, которые, дублируя общие функции, могут иметь диаметрально противоположные подходы к их реализации. Эта ситуация ведет к уменьшению степени управляемости организации в целом, и учебного процесса в частности, что проявляется, например, в десинхронизации результатов и, как следствие, некорректной работе процедур мониторинга и управления учебной организацией. В результате рушится сама стратегия управления такой организацией.

Для решения изложенной проблемы на базе одного из действующих учебных центров университета, был проведен комплекс исследований, целью которых стала выработка основных целей, мер и принципов дефрагментации учебной информационной системы (ИС). В качестве базы будущего дефрагментатора выбран метод взаимной интеграции уже существующих систем с целью обеспечения максимального уровня взаимодействия при сохранении действующих бизнес-процессов внутри каждой системы [1]. Исходя из результатов исследования и последующей разработки технического задания, модуль интегратора должен обеспечить выполнение следующих функций:

- Устранение функциональной избыточности путем создания и сопровождения единой базы данных (БД) для всех систем-участников процесса обучения.

- Поддержку разрабатываемого силами университета анализатора, позволяющего принимать и обрабатывать полученные результаты в реальном времени по множеству необходимых срезов.

- Наличие единой системы мониторинга знаний с целью выработки индивидуальных рекомендаций пользователю на основе его работы во всех системах комплекса.

- Свободный удаленный доступ к БД с целью реализации свободного информационного обмена прогрессом и достижениями.
- Идентификация проприетарных узлов, для которых существует возможность дублирования и замены на свободные (открытые) компоненты.
- Функционирование универсальных методов работы с потоками данных ИС как объектов процесса интеграции.

Полная универсальность такого подхода есть идея химерическая, на практике недостижимая, т.к. невозможно заранее абсолютно точно предсказать полный формат процесса ввода-вывода еще неизвестных методу ИС. Возможность обучения новым принципам форматирования потоков данных заложена на уровне архитектуры команд интегратора и является для него во многом образующей.

Следует отметить, что, согласно одной из задач проекта, в учебную систему предлагается внедрение динамической системы рангов и достижений, являющейся, в основном, прерогативой современных массовых многопользовательских развлекательных онлайн-проектов (ММО). При проведении экспериментального исследовательского запуска единой обучающей системы, такая система достижений с большим отрывом заняла первое место среди мотиваторов процесса обучения. Достижения ранжируются по типу (учебные, информационные и т.п.), виду (теоретические, практические), принадлежности (курсы, дисциплины) и рангу (золото, серебро, бронза). По результатам отметок в данной системе предполагается, что при успешном окончании обучения, выпускнику будут положены некоторые материальные преференции, выдаваемые, например, в качестве премий или памятных знаков, фиксирующих рекордные, наиболее важные и сложные в получении достижения.

Одним из наиболее популярных способов мотивации при взаимодействии с абстрактными системами заслуженно является рейтинговая система, доказавшая свою востребованность во множестве сфер человеческой деятельности. От личного и до государственного рейтинги являются неотъемлемой частью нашей жизни. Работа, учеба, технологии, культура, спорт, финансы... и даже жизнь и война непосредственно завязаны на ту или иную рейтинговую таблицу. Для многих людей, систем и организаций стало самоцелью достижение определенного рейтингового положения. Обобщая, можно считать, что и уровни образования, и организационные должности, и армейские или ученые звания являются локальными рейтинговыми системами.

Возвращаясь к вопросу собственной рейтинговой системы для обучающей и аттестационной ИС, следует отметить модульность. На

данном этапе в виде модулей реализованы следующие компоненты ИС «Пик»:

1. Рейтинговый модуль.
2. Ранговый модуль.
3. Модуль достижений.
4. Модуль званий.
5. Модуль статистики.
6. Модуль аутентификации.

Рейтинговый модуль представляет собой набор внешне примитивных сводных таблиц, состоящих всего из двух полей и содержащих имя пользователя и его числовой рейтинг. Модуль имеет одну общую таблицу и множество разноуровневых локальных. Локальные рейтинговые таблицы привязываются к каждому исполняемому объекту аттестационной ИС, начиная с конкретных периодов и дисциплин, и заканчивая учебными программами и квалификациями. Основным назначением данного модуля является определение точного относительного положения пользователя в учетной группе.

Ранговый модуль используется как средство простого ранжирования пользователей ИС. Основным критерием смены одной ранговой принадлежности пользователя на другую является его личный рейтинг. Сам ранг является символьным (текстовым) обобщением числового рейтинга, лежащего в определенном диапазоне. Для центральной автоматизированной информационно-справочной системы (АИСС) информационного пространства рассматриваемой образовательной организации принято однозначное упорядоченное соответствие текстового ранга числовому коэффициенту, вычисляемому по формуле $R = \text{round}(r/1000)$. Таким образом, на каждую тысячу личного рейтинга назначается свой определенный ранг.

Модуль достижений предназначен для реализации виртуальной наградной системы и в целом идеологически соответствует государственной системе поощрений, связанной с различными благодарностями, грамотами, знаками, медалями, орденами и т.п. Различные виды и типы наград отмечают самые разнообразные вехи во взаимодействии пользователя с АИСС. Так, например, есть специальные награды за самый быстрый верно пройденный тест, освоенный курс, аттестованную программу и т.п. Также существуют награды, явным образом напрямую не связанные с процессом обучения, например, за ежедневное успешное прохождение различных аттестаций в течение определенного периода, прослушивание лекций, идеи и материалы по повышению различных характеристик (удобства, полноты, актуальности, привлекательности, доступности и т.д.) курса, за адекватные примечания и комментарии к самым различным узлам АИСС и т.п.

Достижения классифицируются по способу присвоения (автоматические и модерируемые), по количеству во владении (накопительные, уникальные и глобальные), по времени владения (постоянные и временные), по источнику присвоения (локальные, модульные, системные и сводные), по доступности получения (общего доступа, ограниченного доступа и периодические).

Визуализация многих достижений спроектирована таким образом, чтобы стать частью воплощения системы реальных поощрений организации. В первую очередь это касается уникальных, глобальных и ограниченных достижений. Будучи воплощенными в железе в виде узнаваемых, однозначно идентифицируемых значков, медалей или орденов они становятся мощнейшим мотиватором для плодотворной работы с системой и, как следствие, приобретения соответствующих знаний и навыков.

Система званий является обобщающим фактором, состоящим из двух компонент: собственно, звания и титула. В отличие от ранга, у одного пользователя званий может быть несколько и к каждому такому званию привязан единственный титул, его характеризующий. Звания присваиваются по сумме достижений (обычно сводных или уникальных) и текущему рейтингу и являются по своей сути их производной. Титул обычно характеризует период владения пользователем данным званием.

Назначение званий в целом соответствует таковому для достижений. Принципиальное же отличие званий в том, что их основное воплощение текстовое, в отличие от графического для достижений. Например, во внутренней системе во время официальной переписки или визирования внутренних документов пользователь имеет законное право наряду с должностью, официальным ученым званием и степенью, выбрать одно из доступных ему внутренних званий в качестве части подписи. Таким образом, модуль званий в большей степени ориентирован на сотрудников образовательной организации (преподавателей, администрации, вспомогательного персонала и т.п.).

Подсистема статистики предназначена для оперативной обработки данных других подсистем и модулей рассматриваемой ранговой ИС. Своим функционалом она задействует определенный блок алгоритмов ИС "Анализатор", предназначены для обработки больших массивов эмпирических данных и выявления закономерностей на их основе. Таким образом, становится возможным оперативно отслеживать основные тенденции заинтересованности и развития пользователей внутри ИС в рамках множества срезов, например, по профилям, направлениям, возрасту и специализации, сезонности и т.п.

Для реализации стратегии оптимального управления имеющимися ресурсами интегратором предусмотрены внутренние системные интерфейсы для ИС принятия решений «Каскад», имеющейся в составе

функционирующей АИСС «ВУЗ» как отдельной конфигурации платформы.

На примере интеграции ИС математической статистики «МатСтат» с ИС учета и управления ресурсами «Ресурс» АИСС «ВУЗ» демонстрируются и оцениваются основные элементы, показатели и особенности модуля «Интегратор». Несмотря на свое основное назначение, модуль также является важным функциональным дополнением к компоненте «Анализатор» за счет имеющихся механизмов конвертации сложных структур данных и их представления в удобном для обработки и отображения виде.

Взаимная интеграция ИС и конфигураций средствами условно-универсального модуля, обеспечивающего гибкий интерфейс для каждой клиентской ИС, позволяет воплотить в действительности идею единого информационного пространства организации, включающего в себя помимо системы обучения также кадровые, учетно-складские, бухгалтерские и административные ИС на базе соответствующих программных продуктов. Единое поле обладает рядом несомненных достоинств, таких как:

- Общая, всегда актуальная БД, позволяющая в реальном времени отслеживать динамику протекающих в АИСС процессов, а также экономить целый пласт ресурсов, обычно затрачиваемых на процессы синхронизации данных отдельными БД ИС.
- Центральный портал доступа к сведениям БД, обеспечивающий единый интерфейс для всех категорий пользователей АИСС.
- Унификация средств обеспечения безопасности при поддержке ИС «Анализатор» позволяет максимально оперативно выявить угрозу любому компоненту АИСС и изолировать подвергшийся атаке участок вместе со злоумышленником в специализированной «песочнице».

В АИСС «ВУЗ» интегратор выступает в роли программного коммутатора с функцией шлюза, потому весь его обменный функционал можно рассматривать как модель локальной вычислительной сети и межсетевого взаимодействия [2]. Клиентскими узлами в данной модели выступают объекты работы интегратора, т.е. ИС, подлежащие организации взаимодействия. Роль сервера и контроллера берет на себя образующая АИСС «ВУЗ».

Функционал шлюза в рассматриваемой модели ИС реализован комплексом генетических алгоритмов, реализующих постулаты теории приближений относительно потоков и каналов передачи данных как объектов мутаций отбора [3]. Коммутатор интегратора реализует систему распределенного хранения и передачи данных путем инициализации авторизированных туннелей класса «коммутатор-узел», экземплярами которого являются связи вида «интегратор-система» и «интегратор-

поле», отвечающие, соответственно, за каналы связи коммутатора с клиентскими ИС и корневой АИСС.

Моделирование работы программной топологии с целью выявления ее слабых сторон и явных недостатков возможно провести в большинстве соответствующих программных сред. Например, среда «Трассер», являющаяся частью АИСС «ВУЗ» и используемая для обучения контингента студентов основам работы с корпоративными ИС и различными инфокоммуникационными системами, позволила оценить рассматриваемую модель на предмет надежности, избыточности и ресурсоемкости.

Развитие топологии данной сетевой модели во времени напрямую зависит от этапа внедрения интегратора в действующее информационное поле ВУЗа. Так, на раннем этапе, когда при сохранении существующих прямых связей между различными ИС вида «система-система» и «система-поле» выстраиваются новые связи вида «интегратор-система» и «интегратор-поле», топология является полносвязной, наследуя все ее основные особенности, достоинства и недостатки.

На промежуточном этапе, при частичном сохранении связей «система-система» разрушается большинство связей типа «система-поле», функционал которых полностью заменяется совокупностью связей «интегратор-система» и «интегратор-поле», топология становится сетевой и по структуре напоминает граф, вершиной которого является соответствующий интерфейс ввода-вывода АИСС «ВУЗ».

На заключительном этапе внедрения полностью разрушаются остаточные связи типов «система-система» и «система-поле», тем самым исключая прямое взаимодействие между различными ИС информационного поля. Топология модели взаимодействия программных компонентов трансформируется в полноценную звезду, в центре которой находится интегратор, лучами являются связи типов «интегратор-система» и «интегратор-поле», а вершинами становятся ИС программного комплекса.

В соответствии с современными тенденциями разработки и внедрения различных ИС в организации рассматриваемого типа необходимо обеспечить высочайшую степень защиты информации от всех возможных видов угроз. Несанкционированный доступ к информации такого типа может повлечь за собой как гражданскую и административную, так и уголовную ответственность. Унифицированная централизованная система безопасности, в том числе, максимально полно адаптирована под обеспечение работы таких подразделений как архивы, управления информатизации, учебно-методические управления, управления научно-инновационной деятельности и т.п.

Одним из основных критериев оценки эффективности функционирования АИСС такого класса является показатель

(коэффициент) масштабируемости модели и системы в зависимости от числа клиентских модулей ИС и задействованных систем обеспечения безопасности. В соответствии с проведенными исследованиями при пиковых нагрузках актуальная топология АИСС «ВУЗ» по данному показателю демонстрирует рост в 17-21% по отношению к ее предыдущей версии, что в перспективе обеспечит значительно большую эффективность внедрения новых и модернизацию существующих компонент.

Рассмотренный интегратор позволяет облегчить работу персонала организации всех категорий, в том числе преподавателей, администраторов, инженеров и т.д., а также повысить степень достоверности работы аналитических подразделений, управленческих центров и прогнозирующих служб, что крайне положительно скажется на экономических показателях работы предприятия. В целом, автоматизация процесса интеграции учебных, административных и других ИС позволяет значительно увеличить самые различные коэффициенты эффективности функционирования организации и, как следствие, повысить ее общий рейтинг.

Список использованных источников

1. Журавлев А.Е. Об автоматизации системы контроля знаний в вузе в соответствии с положениями Болонского процесса // Новые информационные технологии в образовании: Сборник научных трудов четырнадцатой международной научно-практической конференции «Применение технологий «1С» для повышения эффективности деятельности организаций образования» 28-29 января 2014 г. Часть 2. М.: ООО «1С-Пабблишинг», 2014. С. 77.

2. Журавлев А.Е. Автоматизация системы оценки качества освоения учебной программы // Материалы XXV международной конференции «Применение новых технологий в образовании», «ИТО-Троицк-2014». М.: БАЙТИК, 2014. С. 412-413.

3. Журавлев А.Е. Особенности взаимодействия различных комплексов информационной экосистемы ВУЗа на базе платформы 1С:Предприятие 8 // Новые информационные технологии в образовании: Сборник научных трудов 16-й международной научно-практической конференции "Новые информационные технологии в образовании" (Применение технологий "1С" в условиях модернизации экономики и образования) 2–3 февраля 2016 г. /Под общ.ред. проф. Чистова Д.В. Часть 2.– М.: ООО "1С-Пабблишинг", 2016. 336 с.: илл. Стр. 54-57.

Сравнительный анализ открытых систем разработки онтологий DOE и PROTÉGÉ

*Конькова Д. С.,
ФГБОУ ВО «Магнитогорский государственный технический
университет им. Г.И. Носова»*

Аннотация. В статье приводится сравнительный анализ программных продуктов Protégé и DOE, которые предназначены для создания, редактирования и просмотра онтологических баз знаний. Выделены критерии сравнения, показано преимущество Protégé.

Ключевые слова: онтология, Protégé, DOE, анализ, сравнение

Идея создания универсальной машины, которая бы решала за людей сложные задачи существовала всегда. С момента, когда появились первые в мире счеты, человечество стремится развивать и изобретать то, что могло бы облегчить жизнь. Создание компьютеров открыло новые горизонты в автоматической обработке информации.

Темпы развития скорости вычисления компьютеров росли, и поэтому встает вопрос: достигнет ли компьютер уровня развития человека? Ответом на данный вопрос занимается такая наука как кибернетика. Кибернетика — наука об общих закономерностях получения, хранения, передачи и преобразования информации в сложных управляющих системах, будь то машины, живые организмы или общество [1]. Сфера применения кибернетики огромна, но если рассматривать ее направления в вычислительной технике, то методы кибернетики применяются для управления устройствами и анализа информации.

Одним из «направлений» кибернетики является искусственный интеллект. Искусственный интеллект (ИИ, англ. AI) — 1) наука и технология создания интеллектуальных машин, особенно интеллектуальных компьютерных программ; 2) свойство интеллектуальных систем выполнять творческие функции, которые традиционно считаются прерогативой человека [5].

Современное развитие искусственного интеллекта можно разделить на два направления:

1) решение проблем, связанных с приближением специализированных систем ИИ к возможностям человека, и их интеграции, которая реализована природой человека;

2) создание искусственного разума, представляющего интеграцию уже созданных систем ИИ в единую систему, способную решать проблемы человечества [4].

Искусственный интеллект вовлекает множество предметных областей, имеющих скорее практическое отношение к ИИ, а не

фундаментальное. Многие подходы были опробованы, но к созданию искусственного разума ни одна исследовательская группа пока так и не подошла.

В искусственном интеллекте также рассматривается вопрос построения онтологий. Интеллектуальными системами называются программы, которые моделируют некоторые стороны интеллектуальной деятельности человека. Понятие онтологии больше философское и означает учение о сущем, либо учение о бытии (в широком смысле существование) как таковом. В информатике же онтология определяется как общий словарь для тех, кому нужно совместно использовать информацию о предметной области [**Ошибка! Неизвестный аргумент ключа.**, 8].

Знания, которые должны быть представлены в онтологиях, должны быть декларативными, то есть должны представлять собой информацию о свойствах и фактах в определенной области. Но знания, которые закладываются в компьютерных программах, бывают еще и процедурными, то есть знания о том, что надо сделать в конкретный момент времени и в конкретной ситуации. Таким образом, при разработке интеллектуальной системы приходится учитывать такое разделение знаний и придумывать специальные программные продукты для оперирования этими знаниями [**Ошибка! Неизвестный аргумент ключа.**, 9].

Построение онтологий – сложный и отнимающий много времени процесс. Первые среды разработки начали появляться в 90-х годах прошлого века. Они предоставляли интерфейсы, которые позволяли выполнять концептуализацию, реализацию, проверку непротиворечивости и документирование.

Сейчас существует множество коммерческих программных продуктов для создания и редактирования онтологий, но также существуют и свободно распространяемые, такие как DOE и Protégé. Рассмотрим сравнительную характеристику каждого из этих продуктов поподробнее.

Protégé – локальная Java-программа, находящаяся в свободном доступе. Была разработана в Стенфордском университете группой медицинской информатики. Интерфейс программы представлен на рисунке 1.

Программа предназначена для создания, редактирования и просмотра онтологий, предназначенных для любой области. Её первоначальная цель – помочь разработчикам программного обеспечения в создании и поддержке явных моделей предметной области и включение этих моделей непосредственно в программный код.

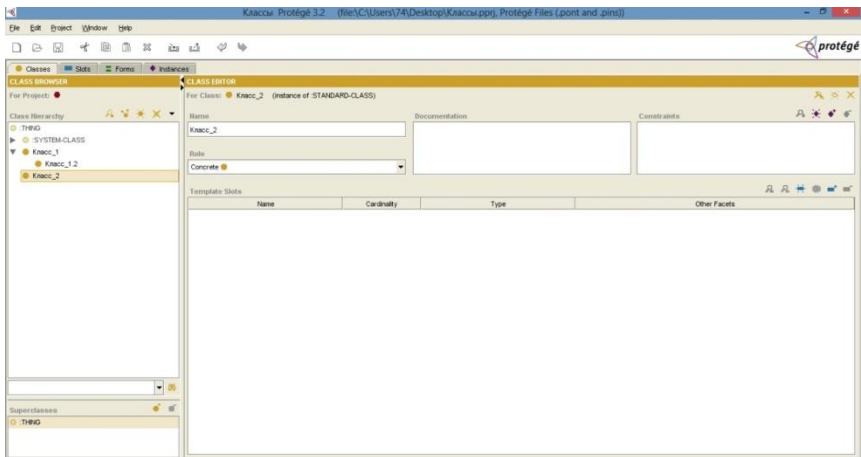


Рис.13. Интерфейс Protégé

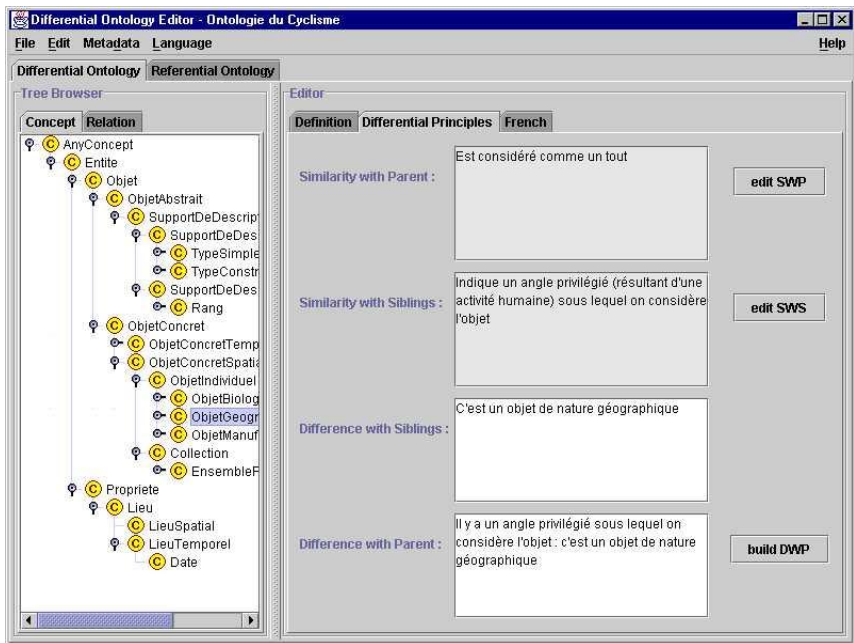


Рис.14. Интерфейс DOE

Структура онтологии сделана аналогично иерархической структуре каталога. На основе сформированной онтологии, Protégé может генерировать формы получения знаний для введения экземпляров

классов и подклассов. Она включает редактор онтологий, позволяющий проектировать онтологии, разворачивая иерархическую структуру абстрактных и конкретных классов и слотов. На основе сформированной онтологии Protege позволяет генерировать формы получения знаний для введения экземпляров классов и подклассов [3].

DOE, в отличие от Protégé, - простой редактор онтологий, который позволяет пользователю строить базы знаний в соответствии с методологией, предложенной Бруно Бачимонта. Интерфейс программного средства представлен на рисунке 2.

Процесс построения онтологии разделен на 3 этапа. На первом этапе пользователю предлагается создать таксономию понятий и отношений, описывая как иерархию. Кроме того, пользователь может добавлять синонимы и энциклопедические определения для всех понятий. На втором этапе, таксономии рассматриваются с экстенциональной семантической точки зрения. Пользователь может дополнить их новыми объектами (определенными) или добавить ограничения на домены отношений. Наконец, в третьей стадии, онтология может быть переведена на язык представления знаний, что позволяет импортировать его в другой инструмент онтологий.

Для более наглядного сравнения этих двух программ, была составлена Таблица 3, в которой по наиболее важным критериям проводился анализ систем.

Таблица 3

Сравнительная таблица программ DOE и Protégé

Критерии сравнения	Protégé	DOE
Специфика структуры редакторов	классы, слоты (=свойство), экземпляры, фасеты (= ограничения слотов).	класс, свойство, домен
Экспорт данных	в RDF, RDFS, XML, HTML, OWL, Clips, N3, Turtle.	RDFS, COBA, DAML + OIL (март 2001 г.), OIL, CGXML, язык для определения концептуальных графов.
Запросы	сложные запросы в рамках онтологии: выборка, проверка и т.д.	нет
Язык для представления	существует поддержка русского языка для представления	нет поддержки русского языка для представления
Базовый язык	OKBC model	XML & CGXML
Поддержка Web-совместимых онтологий и	ограниченно пространство имени и может работать как	поддерживает URL

Критерии сравнения	Protégé	DOE
использование ПО через Интернет	апплет, доступ через сервлеты	
Графическое представление	просмотр классов и глобальных свойств с помощью плагина GraphViz, вложенный просмотр графов с помощью редактирования плагина Jambalaya	нет, существует представление в виде дерева
Проверка согласованности	плагины для добавления и проверки ограничений аксиом: PAL, FaCT.	обнаруживает циклы в иерархии
Многопользовательская поддержка	в стадии разработки на версии Protégé 2000	нет
Лексическая поддержка, возможности для лексического привязки онтологий элементов и обработки лексического содержания (например, поиск / фильтрация терминов)	WordNet плагин, подстановочные соответствия строк (только API)	определение терминов, синонимов и предпочтений, методология дифференциальных определений
Дополнительные функции	поддержка методологии CommonKADS	поддержка методологии Б.Бачимонта, которая будет использоваться с другими редакторами

Два этих программных продукта очень схожи, но в плане наполненности, функциональности и ориентированности на русскоязычного пользователя, Protégé обходит DOE. Так как DOE не используется в качестве полной среды разработки онтологий, он не будет активно поддерживать множество возможностей, которые традиционно присутствуют в онтологических редакторах. Protégé привлекательней уже только потому, что в нем можно создавать как фреймовые, так и семантические представления и так же не маловажны запросы, которых в DOE попросту нет.

Нельзя однозначно сказать какое средство лучше, а какое хуже, но наверняка можно сказать то, что Protégé – это более полная программа для построения онтологий, чем DOE.

Список использованных источников

1. «Энциклопедия кибернетики» под ред. В. М. Глушкова, т.1., Киев, 1974 — с. 440.
2. Ontology editor survey results [Электронный ресурс]. – Режим доступа: http://www.xml.com/2002/11/06/Ontology_Editor_Survey.html
3. Protege [Электронный ресурс]. – Режим доступа: <http://protege.stanford.edu/about.php>
4. Tadviser. Государство. Бизнес. ИТ. [Электронный ресурс]. – Режим доступа: http://ww.tadviser.ru/index.php/Продукт:Искусственный_интеллект
5. Аверкин А. Н., Гаазе-Рапопорт М. Г., Поспелов Д. А. Толковый словарь по искусственному интеллекту. — М.:Радио и связь, 1992. — 256 с.
6. Белоусова И.Д. К вопросу о согласовании требований к содержанию профессиональной подготовки на основе онтологической модели / И.Д. Белоусова, Л.В. Курзаева, А.М. Агдавлетова // Современные наукоемкие технологии. - 2015. - № 11. - С. 67-70.
7. Белоусова И.Д., Курзаева Л.В., Лактионова Ю.С., Агдавлетова А.М. Онтологическая модель управления требованиями в процессе подготовки ИТ-специалистов // Успехи современной науки. 2016. Т. 1. № 3. С. 98-100.
8. Курзаева Л.В. Дистанционный курс «Инструментальные методы поддержки принятия решений»: электронный учебно-методический комплекс // Хроники объединенного фонда электронных ресурсов Наука и образование. 2016. № 1 (80). С. 2.
9. Курзаева, Л.В. Введение в теорию систем и системный анализ: учеб. пособие / Л.В. Курзаева. -Магнитогорск: МГТУ им. Г.И.Носова, 2015. -211 с.

Построение онтологии «Автодилеры» в программной среде Protégé 4.2

*Сапегина В. С.,
ФГБОУ ВО «Магнитогорский государственный технический
университет им Г.И. Носова»*

Аннотация. Целью работы являлась разработка онтологии «Автодилеры» в программной среде Protégé 4.2. Была построена иерархия классов, разработаны атрибуты и свойства экземпляров. Планируется использовать созданную онтологию в качестве основы электронной энциклопедии по автодилерам.

Ключевые слова: онтология, Protégé, класс, иерархия, разработка, автодилеры.

Переход к информационному обществу оказал значительное влияние на мировую экономику, которая опирается на информационно-коммуникативные технологии (ИКТ), которые создают удобную и производительную среду для совместной работы. В то же время их возможности не исчерпываются только упрощением взаимодействия между бизнес-партнерами. Используя достижения в области искусственного интеллекта, ИКТ позволяют создавать и тиражировать базы знаний, которые являются результатом накопления специфического опыта работы всех экспертов в какой-либо конкретной области. Для такой сложной предметной области, как автомобильная промышленность, подобные базы особенно востребованы.

В литературе по искусственному интеллекту [1,2] приводят несколько определений понятия онтологии, при этом некоторые из них противоречат друг другу. В данной статье под онтологией будем понимать формальное явное описание понятий в рассматриваемой предметной области (классов), свойств каждого понятия, описывающих различные свойства и атрибуты понятия (ролями или свойствами), и ограничений, наложенных на слоты (фацетов, иногда их называют ограничениями ролей). Онтология вместе с набором индивидуальных экземпляров классов образует базу знаний.

В настоящее время в онтологии вытупают одной из моделей представления знаний о реальном мире или его части. Построение онтологий – трудоемкий и длительный процесс. Для того, чтобы его облегчить, в середине 90-х годов начали создавать специализированные среды разработки онтологий [4].

В настоящее время для создания и поддержки онтологий существует ряд инструментов, которые кроме основных функций редактирования и просмотра элементов онтологии осуществляют поддержку документирования, импорт и экспорт онтологий в разных форматах и созданных на разных языках описания знаний, поддержку визуального редактирования, управления библиотеками и т.д.

Наиболее известными инструментами разработки онтологий являются: система Ontolingua, Protégé, OntoEdit, OilEd, WebOnto, OntoSaurus, KADS22.В данной работесоздание онтологии будет выполнено в среде Protégé 4.2 [3].

В Protégé 4.2 первым этапом разработки онтологии является создание классов, которое осуществляется в закладке «Classes». Изначально пустая форма содержит один класс с именем THING, представляющий включающий все объекты предметной области набор (3). В предметной области «Автодилеры» были выделены три основных

класса: «Страны», «Классы авто» и «Продажа автомобилей», разбивающийся на подклассы известных марок автомобилей.

На втором этапе разработки онтологии определяют атрибуты класса и свойства экземпляров. В Protégé 4.2 существует два основных типа свойств: свойства объекта (Object Properties) и свойства типов данных (Data Properties). Свойствами объекта являются отношения между двумя экземплярами.

Для классов онтологии были созданы следующие свойства объектов (в скобках указаны классы, чьи экземпляры будут обладать перечисленными свойствами):

- имеет такой же класс как (все подклассы класса «Автодилеры»);
- относится к классу («Классы авто»);
- по качеству производства (все подклассы класса «Автодилеры»);
- производится в («Страны»);
- по расходу топлива (все подклассы класса «Автодилеры»);
- имеет схожесть (все подклассы класса «Автодилеры»).

Окончательный список свойств объектов представлен на рисунке

1.

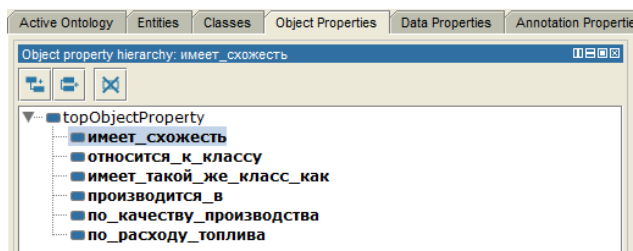


Рис. 1. Список свойств объектов

На третьем этапе определяются свойства типов данных (атрибуты класса), которые описывают связи между индивидом и значения данных. Для создания свойств типов данных в Protégé 4.2 необходимо перейти в закладку «Data Properties». Процесс создания аналогичен созданию классов и свойств объекта. Для классов онтологии будут созданы следующие свойства типов данных (см. рисунок 2):

- 1) количество пассажирских мест;
- 2) количество цилиндров;
- 3) коробка переключения передач;
- 4) объем двигателя;
- 5) привод;
- 6) разгон до 100км/ч;
- 7) расположение руля;

8) расход топлива на 100 км/ч.

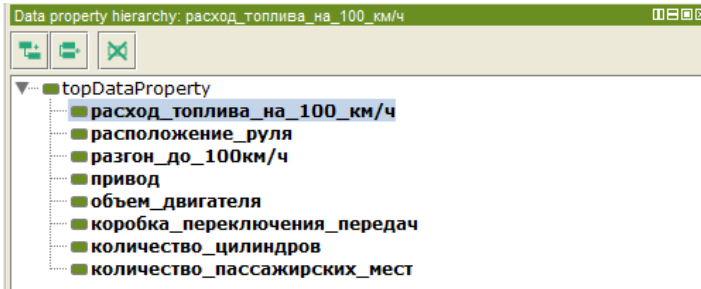


Рис. 2. Свойства типов данных

Предпоследним этапом разработки онтологии является заполнение классов экземплярами. Для добавления экземпляров в Protégé 4.2 необходимо перейти в закладку «Individuals», нажать на иконку add individuals и задать имя экземпляра. После создания экземпляров, необходимо задать свойства объектов между экземплярами классов.

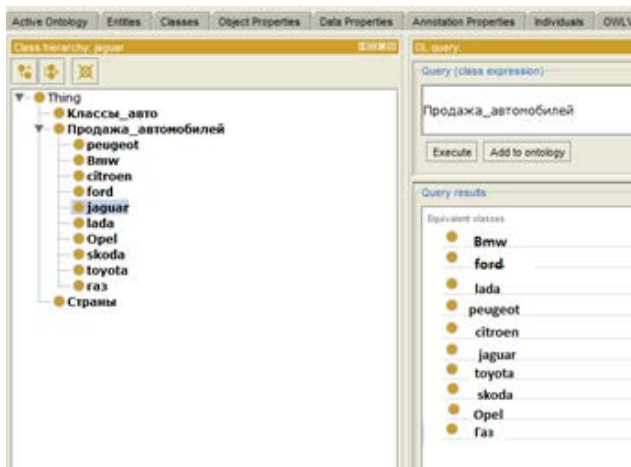


Рис. 3. Поиск подклассов класса «Продажа автомобилей»

Последним этапом разработки выступает создание базовых DL-Query запросов. Они отразят результаты работы, а также могут помочь с поиском необходимой информации в заполненной онтологии.

Для создания запросов необходимо перейти в закладку DL-Query, а затем на вкладке «Reasoner» выбрать «Start reasoner». Например, нужно

вывести все подклассы класса «Продажа автомобилей». Для этого в запросном окне необходимо написать названия класса и выбрать на панели объектов «Descendant classes» (см. рисунок 3).

Итак, разработанная онтология «Автодилеры» содержит 11 классов и подклассов, 6 свойств объектов, 8 атрибутов класса и 17 экземпляров. Был составлен запрос, показывающий работу онтологии. Данную онтологию в дальнейшем можно расширять, добавлять нерассмотренные разделы предметной области и новые связи в случае, если у экземпляров появились новые характеристики, либо добавились объекты, которые также необходимо связать между собой или с другими объектами, а также использовать как основу для создания электронной энциклопедии по автодилерам.

Список использованных источников

1. Гаврилова Т.А. Формирование прикладных онтологий / Гаврилова Т.А. // Десятая национальная конф. по искусственному интеллекту с международным участием КИИ-2006. –М.: Физматлит, 2006.
2. Муромцев Д.И. Онтологический инжиниринг знаний в системе Protege: методическое пособие / Д.И. Муромцев - СПб:СПб ГУ ИТМО, 2011. - 62 с.
3. Овдей О.М. Обзор инструментов инженерии онтологий / О.М. Овдей, Г.Ю. Проскудина. – Режим доступа: <http://www.interface.ru/home.asp?artId=33246>
4. Попова И.В. Технологии поддержки принятия решений / И.В. Попова, Г.А. Лисьев. –М.: Флинта, 2011. – 133 с.

Обзор открытых оболочек для создания экспертных систем

*Столяров А. И., Севостьянова А. В.,
ФГБОУ ВО «Магнитогорский технический университет им. Г.И.
Носова»*

Аннотация. В статье приводится обзор современных свободно распространяемых оболочек экспертных систем, выделены поддерживаемые ими модели знаний и области применения экспертных систем.

Ключевые слова: оболочка экспертной системы, свободное программное обеспечение, Малая Экспертная Система 2.0, COMDALE/C, PROCESSVision, CLIPS, MIKE

На сегодняшний день актуальность использования и разработки открытых экспертных систем обусловлена проблемой формирования единого информационного пространства, необходимого для решения многих социально-экономических задач в процессе становления и развития информационного общества. Распространение информационных систем и технологий, вычислительной и телекоммуникационной техники в различные сферы человеческой деятельности привело к тому, что стало необходимо объединить вычислительные устройства и реализованные на их основе информационные системы в единые информационно-вычислительные системы и среды.

Наиболее важным достоинством экспертных систем является возможность накопления экспертных знаний и хранение их длительное время. Анализ любой информации экспертные системы выполняют объективно, что улучшает качество проводимой экспертизы [1].

Существует огромное количество определений понятия «открытая система», сформулированных в отдельных крупных компаниях и компаниях по стандартизации.

По мнению специалистов Ассоциации французских пользователей UNIX и открытых систем (AFUU), «открытой является система, состоящая из элементов, которые взаимодействуют друг с другом через стандартные интерфейсы» [7].

Программная или информационная система должна иметь совокупность следующих свойств, для того, чтобы её можно было отнести к открытой системе:

- интероперабельность – это способность к взаимодействию с другими прикладными системами на локальных и удаленных платформах;

- стандартизуемость – программные и информационные системы проектируются и разрабатываются на основе согласованных международных предложений и стандартов, на базе функциональных профилей в области информационных технологий осуществляется реализация открытости;

- масштабируемость – это возможность перемещения прикладных программ и передачи данных в системах и средах, обладающих разными характеристиками производительности и разными функциональными возможностями, возможность создания дополнительных функций информационной системы или модификации некоторых уже имеющихся при неизменяемых остальных функциональных частях информационной системы;

- мобильность – обеспечение возможности переноса прикладных программ и данных при модернизации или замене аппаратных платформ

информационной системы и возможности работы с ними специалистов, использующих информационные технологии, без их специальной переподготовки при модификации информационной системы;

– дружелюбность к пользователю – развитые унифицированные интерфейсы в процессах взаимодействия в системе «пользователь – ПК – ПО», позволяющие работать пользователю, не имеющему специальной системной подготовки [8].

Стоит отметить, что отдельные свойства открытых систем характерны для любой автоматизированной информационной системы. Современный взгляд на открытые системы состоит в том, что эти характеристики рассматриваются и используются в комплексе – как взаимосвязанные и реализующиеся в целом. Именно в таком комплексе возможности открытых систем позволяют в достаточной степени решать проблемы проектирования, разработки, внедрения, эксплуатации и развития современных информационных систем.

В рамках данной статьи была предпринята попытка рассмотреть и проанализировать различные открытые оболочки для создания экспертных систем.

Рассмотрим наиболее распространенные оболочки для создания экспертных систем: Малая Экспертная Система 2.0, COMDALE/C и COMDALE/X, PROCESS Vision, CLIPS, FuzzyCLIPS 6.02, MIKE.

Малая Экспертная Система 2.0

Эта программа представляет собой простую оболочку экспертной системы на основе байесовской системы логического вывода. МЭС предназначена для решения задач пользователя в какой-либо прикладной области для определения вероятностей возможных исходов, используя для этих целей оценку правдоподобности некоторых предпосылок, которые система получает от пользователя.

В комплекте с системой имеются примеры баз знаний, например, медицинская база знаний по 89 болезням и база для идентификации микроорганизмов.

Система выступает в роли эксперта, который задаёт человеку вопросы и на основе полученных сведений выводит решение. Причём, желательно задавать наиболее значимые вопросы, от ответа на которые в большей степени зависит окончательное решение. МЭС запрашивает у пользователя оценку истинности наиболее важного свидетельства, на основе ответа корректирует вероятности исходов и переходит к следующему свидетельству, выбрав снова самое актуальное. Данный подход позволяет достичь наискорейшего получения результата при минимальном количестве запросов.

Применение байесовской системы логического вывода говорит о том, что информация, обрабатываемая данной экспертной системой, не является совершенно точной, а скорее, носит вероятностный характер.

Пользователь не может быть уверен в полной истинности или ложности ответов, он может отвечать на запросы системы с некоторой степенью уверенности. Система же выдаёт результаты консультации в виде вероятностей наступления исходов.

Важным достоинством программы «Малая Экспертная Система» можно назвать возможность создания, редактирования и использования собственной базы знаний. Чтобы облегчить данную задачу, был написан Редактор баз знаний 1.0, поставляемый вместе с системой.

Возможности редактора:

- коррекция файлов больше 64 килобайт;
- показ текущих координат курсора;
- поиск и замена частей текста;
- возможность проверки созданной базы знаний на ошибки без запуска «Малой Экспертной Системы»;
- работа паролями баз знаний [6].

COMDALE/C и COMDALE/X

COMDALE/C – это экспертная система реального времени, созданная для мониторинга процессов в условиях производства. Система позволяет формировать рекомендации и заключения в рамках непрерывного процесса принятия решения. Система способна обрабатывать неопределенные знания и данные, и имеет открытую архитектуру. Особенности системы:

- объектно-ориентированная конфигурация;
- возможность организации работы в сети;
- обработка прерываний;
- хранение и обработка данных.

COMDALE/X – это консалтинговая экспертная система, позволяющая работать в режиме реального времени. Данная система используется как инструмент разработки экспертных систем реального времени в сочетании с системой COMDALE/C. Она предоставляет возможность включить в экспертную систему гипертекст, что в свою очередь позволяет создавать справочники с удобным интерфейсом [2].

PROCESS Vision

PROCESS Vision– это пакет программ, позволяющий управлять процессами в реальном времени. В его основе лежит открытая и модульная архитектура.

В пакет ProcessVision входит графический интерфейс оператора и объектно-ориентированный дисплей. Система способна выполнять проверку правильности показаний датчиков и поддерживать связь с производственной контрольно-измерительной аппаратурой в единой вычислительной среде [2, 8].

CLIPS

Платформа CLIPS предоставляет возможность быстрого создания эффективных, компактных и легко управляемых экспертных систем. Пользователь может применять множество уже готовых инструментов: так, в систему включены механизмы управления базой знаний и логического вывода, а так же менеджеры различных объектов и конструкций.

К тому же CLIPS предоставляет возможность разделения базы данных и решения задачи на отдельные независимые модули. Экспертная система CLIPS может применяться в нечётких системах для формализации знаний.

Потомками CLIPS являются следующие языки и среды:

- Jess – коммерческая оболочка, доступная бесплатно для академического использования, переписана на Java, поддерживает обратный вывод.

- HaleyRules Eclipse – язык правил системы HaleyRules компании Haley Systems. Поддерживает обратный вывод.

- FuzzyCLIPS.

- EHSIS – испанский CLIPS, разработан в группе Etabaki Факультета Информатики Университета Страны Басков. Поддерживает нечеткую логику как FuzzyCLIPS.

- CLIPS/R2 – оболочка, разработанная в компании Production Systems Technologies, обратно совместимая со стандартным CLIPS версий 6.x, но с добавлением алгоритма Rete II, поддержкой обратного вывода и прозрачной поддержкой структур языка Си и классов C++.

К основным минусам системы можно отнести тот факт, что существующие версии CLIPS для Windows не поддерживают кириллицу. Отсутствие полноценной локализации выступает основным препятствием для свободного распространения CLIPS в России. В то же время открытость исходных кодов CLIPS позволяет исправить эту ситуацию.

Последняя версия – CLIPS 6.3 Beta, выпущена 7 апреля 2008 [3].

FuzzyCLIPS 6.02

FuzzyCLIPS 6.02 – это основанная на правилах версия CLIPS, которая используется для представления и управления нечеткими фактами и правилами. В дополнение к функциональным возможностям CLIPS, FuzzyCLIPS работает с точными, нечеткими знаниями, сложными рассуждениями, которые смешиваются в правилах и фактах экспертной системы. Имеются версии для систем UNIX, Macintosh и IBM PC. Программное обеспечение распространяется бесплатно, но документация по FuzzyCLIPS имеет сроки использования [3].

MIKE

MIKE (Микро Интерпретатор для инженерии знаний) – это открытая программная среда, разработанная для обучения технологии

экспертных систем в Открытом Университете Великобритании. Система базируется на прямых и обратных правилах вывода от цели к фактам, на определяемых пользователем стратегиями разрешения противоречий, и фреймовом языке представления знаний, поддерживающим механизм наследования. Правила вывода автоматически поясняются, пользователь имеет возможность сформировать объяснения «почему». Порядок использования правил в процессе трассировки и работы может отображаться графически на дисплее. MIKE написан на одном из диалектов Пролога, исходный текст программы свободно распространяется [4,9].

По результатам анализа была составлена сравнительная таблица.

Таблица 1

Сравнительная таблица оболочек экспертных систем

Название	Модель данных	Область применения
Малая Экспертная Система 2.0	Байесовская система логического вывода	Множество
COMDALE/C и COMDALE/X	Объектно-производственная	Процессы производства
PROCESS Vision	Объектно-производственная	Процессы производства
CLIPS	Логический вывод от фактов к цели	Множество
FuzzyCLIPS 6.02	Нечеткие правила и факты	Множество
MIKE	Прямые и обратные правила вывода	Множество

В результате была выбрана Малая Экспертная Система 2.0, которая имеет наиболее интуитивно понятный русифицированный интерфейс, большую область применения и грамотную русскоязычную документацию, поставляемую с системой. Оболочка может быть рекомендована для построения экспертных систем различной направленности.

Список использованных источников

1. Гаврилова И.В., Масленникова О.Е. Основы искусственного интеллекта / И. В. Гаврилова, О.Е. Масленникова. – М.: Флинта, 2013. – 282 с.
2. Оболочки для создания Экспертных систем [Электронный ресурс] // Bourabai Research. - Режим доступа: <http://bourabai.ru/alg/expert22.htm> (дата обращения 26.04.2016)

3. Новикова В.А., Андреева Е.Ю., Туйкина Д.К., Искусственный интеллект и экспертные системы [Электронный ресурс] // Экспертные системы. Режим доступа: http://expro.ksu.ru/materials/ii_i_es/book.html (дата обращения 26.04.2016)

4. Каширина И.Л. Нейросетевые технологии: учебное пособие / Каширина И.Л. - Воронеж:Издательско-полиграфический центр ВГУ,2008г.-72

5. Жариков О.Г., Литвин А.А., Ковалёв В.А. Экспертные системы в медицине/ Жариков О.Г.//Медицинские Новости.-2008.-№10

6. «Малая экспертная система 2.0 Редактор баз знаний 1.0», Руководство пользователя, Национальный Технический Университет Украины «Киевский Политехнический Институт», Кафедра КЭВА, 2010 г.

7. Понятие открытых систем [Электронный ресурс] // Журнал информационных технологий «IT-технологии». Режим доступа: www.irkinfo.ru/ponyatie-otkrytykh-sistem.html (дата обращения 26.04.2016)

8. Попова И.В. Технологии поддержки принятия решений / И.В. Попова, Г.А. Лисьев. – М.: Флинта, 2011. – 133 с.

9. Гаврилова И.В., Кортикова О.М., Озерова М.И. Применение открытой среды SWI-Prolog в обучении логическому программированию / И.В. Гаврилова, О.М. Кортикова, М.И. Озерова // Теория и практика применения свободного программного обеспечения. – Магнитогорск, МаГУ., 2011. С. 54-61

Создание онтологии «Лекарственные средства» в системе Protege

*Столяров А. И., Севостьянова А. В.
ФГБОУ ВО «Магнитогорский технический университет им. Г.И.
Носова»*

Аннотация. В статье описывается создание онтологии «Лекарственные средства» в среде Protégé. Разработанная онтология представляет собой электронный справочник по лекарственным препаратам, в котором можно легко находить аналоги.

Ключевые слова: онтологии, лекарственные средства, Protégé.

В настоящее время онтологии широко используются в программировании, обучении, различного рода исследованиях. Большое значение они приобретают в такой области как лекарственные средства, их представление в виде схемы классов, связанных между собой посредством различных отношений и правил. Это своеобразная форма представления некоторой области знаний в формальном виде. Лекарственные препараты имеют ярко выраженную классификацию, информация имеет четкую раздробленность и большое множество

данных, которые возможно представить в виде области знаний в системе Protégé [3,4,5].

Под лекарственными средствами понимают вещества или их комбинации, которые:

- вступают в контакт с организмом человека или животного;
- проникают в органы, ткани организма человека или животного;
- применяются для профилактики, диагностики, лечения заболевания, реабилитации, для сохранения, предотвращения или прерывания беременности:

- получены из крови, плазмы крови, из органов, тканей организма человека или животного, растений, минералов с помощью синтеза или применения биологических технологий. К лекарственным средствам относят фармацевтические субстанции и лекарственные препараты.

В предметной области «Лекарственные средства» были определены следующие классы и их подклассы: основной класс «Thing», имеющий подклассы в виде назначения медикаментов и их состава. Кроме этого определены классы «Медикаменты», «Назначение», «Состав», включающий подклассы «Активные вещества» и «Вспомогательные вещества»[1,2]. В результате работы получена иерархия классов (см.рисунок 1).

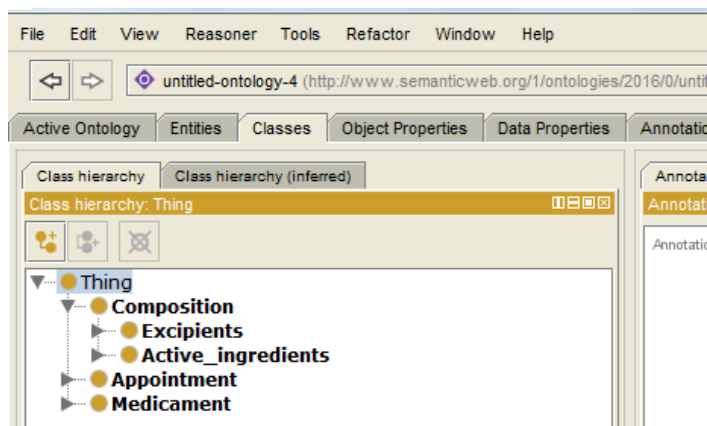


Рис. 1. Иерархия классов

Нагляднее полученную иерархию можно отразить на закладке «OntoGraf».

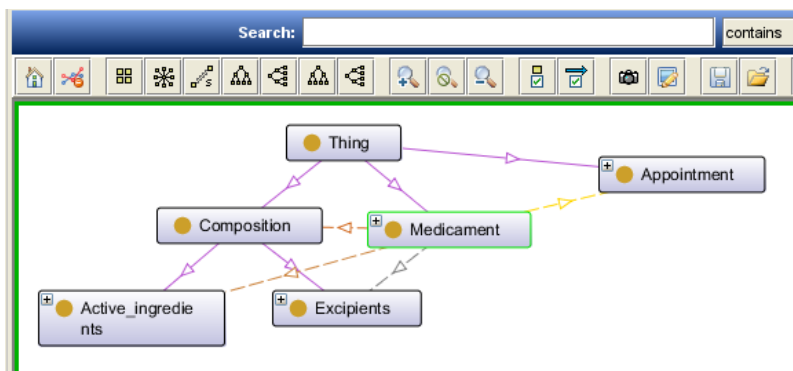


Рис. 2. Дерево классов

Создадим иерархию классов.

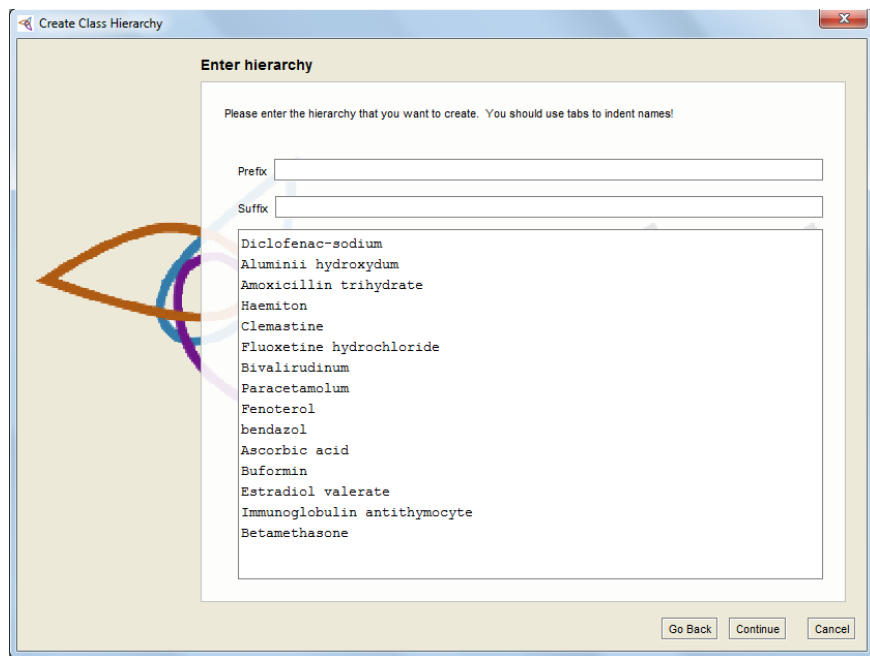


Рис. 3. Создание иерархии классов

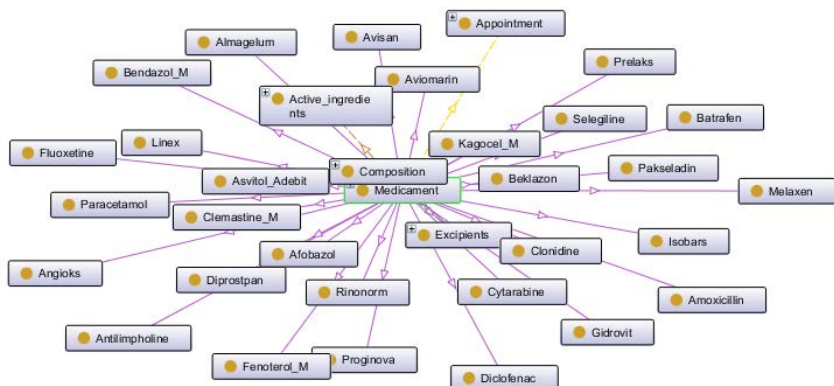


Рис. 4. Дерево иерархии классов

В Protege 4.2 существует два основных типа свойств: свойства объекта и типов данных. В среде по умолчанию создаются свойства объектов верхнего уровня.

Свойства объекта создаются подобным способом, как и классы.

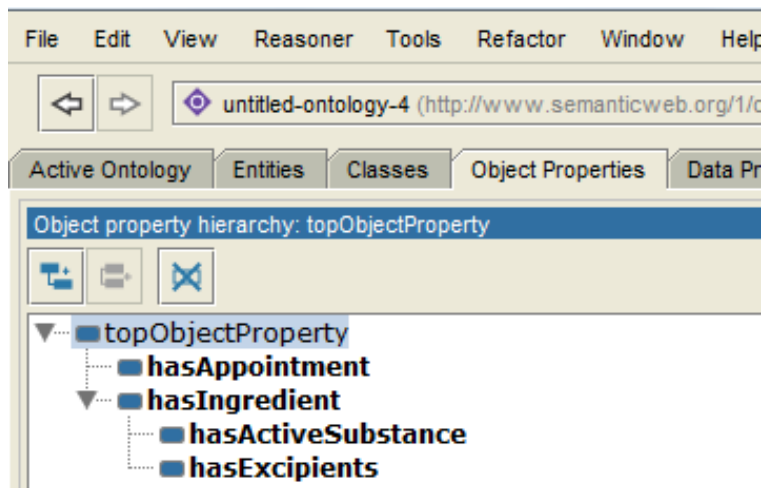


Рис. 5. Созданные свойства объектов

Для создания экземпляров в Protege 4.2 необходимо перейти в закладку «Individuals», нажать на иконку «add individuals» и задать имя экземпляра.

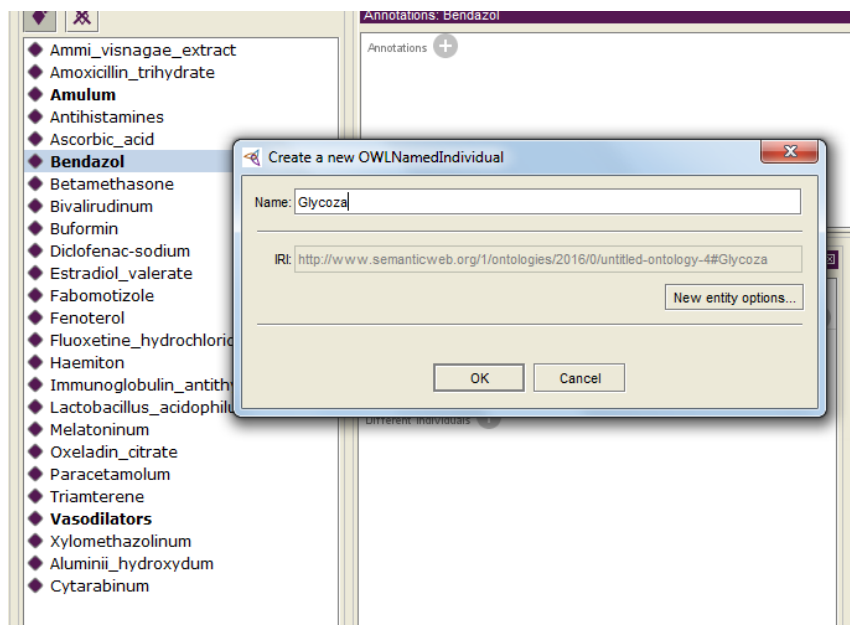


Рис. 6. Создание экземпляра

В рамках данной работы было создано 99 экземпляров, которые распределены по классам следующим образом: Классы медикаменты (30 экземпляров), Состав (Вспомогательные вещества (9 экземпляров), активные вещества (30 экземпляров), Назначения (30 экземпляров).

Процесс создания онтологии лекарственных препаратов, которая описывает различные лекарственные препараты, основанный на их составе и действии, можно считать успешной.

В ходе работы была создана онтология «Лекарственные средства». В нее входят 5 классов и подклассов, 4 типа свойств объектов и 95 экземпляров. Данную онтологию в дальнейшем можно использовать в качестве основы электронной энциклопедии по фармакологии. Онтология предполагает развитие. Планируется создание взаимосвязи лекарственных веществ в рамках онтологии и возможности проверки совместимости лекарственных препаратов, которая имеет большое значение в фармакологии. Данная онтология может предоставить основную фармакологическую информацию.

Список использованных источников

1. Ontology Development 101: A Guide to creating Your First Ontology: практический справочник / Natalya F. Noy, Deborah L. McGuinness - Stanford University: 2001. - 25 с.
2. Гаврилова И.В., Кортикова О.М., Озерова М.И. Применение открытой среды SWI-Prolog в обучении логическому программированию / И.В. Гаврилова, О.М. Кортикова, М.И. Озерова // Теория и практика применения свободного программного обеспечения. – Магнитогорск, МаГУ, 2011. С. 54-61
3. Гаврилова И.В., Масленникова О.Е. Основы искусственного интеллекта / И. В. Гаврилова, О.Е. Масленникова. – М.: Флинта, 2013. – 282 с.
4. Онтологический инжиниринг знаний в системе Protege: / Д.И. Муромцев - СПб:СПб ГУ ИТМО: 2007. - 62 с.
5. Попова И.В. Технологии поддержки принятия решений / И.В. Попова, Г.А. Лисьев. – М.: Флинта, 2011. – 133 с.

Технологическая платформа RadixWare

*Капцан А.В.,
главный конструктор CompassPlus, к. ф.-м. н.
Черномырдин А.В.,
ведущий инженер-программист CompassPlus
Матвеев В., Чернов В.,
ЧОУ «Информационные технологии финансовой индустрии»*

Аннотация. В статье приводится описание платформы RadixWare, разработанной в компании *CompassPlus*. RadixWare представляет собой открытое средство разработки корпоративных информационных систем. Перечислены основные технические характеристики среды, указываются отличия платформы от аналогичных систем. Приводится перечень компонентов RadixWare.

Ключевые слова: RadixWare, корпоративные информационные системы, model-driven development, Java, OpenSource-проект

Технологическая платформа RadixWare является разработкой *CompassPlus*, с мая 2015 года имеющей статус общедоступной и развиваемой в рамках OpenSource-проекта. RadixWare доступна по ссылке <http://www.radixware.org> . Основное назначение платформы –

поддержка разработки информационных систем корпоративного уровня, таких как процессинговые, фронт- и бэк-офисные системы банков, ERP, MES, CRM и т. д.

Технологическая платформа RadixWare (далее Radix) представляет собой, в некотором роде, квинтэссенцию многолетнего опыта *CompassPlus* в области разработки масштабных корпоративных информационных систем. В настоящее время Radix является основным инструментом разработки всех новых программных продуктов и прикладных платформ компании *CompassPlus*, а также основой для перевода ранее разработанных программных продуктов на современную технологическую базу. Попытаемся сейчас кратко обозначить наиболее важные аспекты этой платформы.

Radix изначально разрабатывался как максимально полная среда, содержащая в себе все необходимые средства разработки в области корпоративных информационных систем (далее КИС), с минимальным привлечением в нее компонент сторонних разработчиков. Столь жесткое требование связано, в частности, с тем, что корпоративные информационные системы имеют типичный жизненный цикл, измеряемый десятилетиями. Включение в Radix сторонних компонент вносит в систему в этих условиях большой элемент риска, поскольку заранее неизвестно, в каком направлении будет происходить эволюция этих компонент и будут ли они даже существовать через достаточно длительное время (8–10 лет). Для систем со столь продолжительным жизненным циклом полнота средств разработки является гарантией надежности и работоспособности проекта на все время своего существования.

Основой любого проекта, разрабатываемого в Radix, является *модель*, т. е. платформа поддерживает методологию MDD (model-driven development). Модель системы содержит описание всей системы, начиная от структуры базы данных, описания бизнес-логики, протоколов взаимодействия с внешними системами и заканчивая представлением данных на экране компьютера пользователя. Чрезвычайно важным отличием Radix от подавляющего большинства других систем, реализующих парадигму MDD, является то, что никаких других форм хранения описания проекта, кроме модели, в системе просто нет. Типичные MDD-системы, как правило, позволяют разработать для проекта ту или иную модель, чтобы затем преобразовать ее в некоторое промежуточное представление (например, набор модулей на языках Java, C#, набор SQL-скриптов и т. д.). После этого разработчик может вносить в это промежуточное представление произвольные правки (которые в

некоторых MDD-системах могут быть с теми или иными усилиями включены обратно в исходную модель), но с этого момента и модель, и ее промежуточные представления начинают развиваться независимо друг от друга. Модель проекта в этом случае по факту оказывается *одноразовой*, чем нивелируются все преимущества MDD (непротиворечивость, целостность, возможность анализа и исследования модели и т. д.). В Radix модель проекта преобразуется непосредственно в выполнимый код, и более того, она доступна этому коду на стадии выполнения. Можно утверждать, что реализация MDD в Radix доведена до своего логического конца.

Для описания модели проекта в Radix используется преимущественно декларативный подход. Такое решение позволяет в большинстве случаев избавиться от необходимости разработки программного кода, заменив его описанием компонент модели при помощи диалоговых средств. С учетом того, что время разработки крупных КИС измеряется сотнями и тысячами человеко-лет, применение декларативного подхода существенным образом сокращает время разработки проекта и избавляет последний от многих ошибок. Тем не менее, не исключено применение классических объектно-ориентированных и процедурных средств разработки, поскольку у каждого из подходов есть та область, в которой его использование оказывается более эффективным и рациональным. Основным языком процедурной разработки в Radix является язык JML. Это расширение языка Java, в котором есть возможность употреблять ссылки на элементы модели системы непосредственно в программном коде. Также в Radix поддерживается и язык SQML — расширения языка SQL и PL/SQL, в которых возможен доступ к элементам модели системы. Особо следует остановиться на том, что в Radix на основании модели системы обеспечивается *автоматическое* создание и поддержание в актуальном состоянии всех средств взаимодействия с пользователем (экранных форм, меню, печатных документов) и доступ к ним — как через специальный «толстый» клиент Radix (т. н. RadixWareDesktopExplorer), так и через обычный интернет-браузер. В тех случаях, когда использование автоматически созданных средств взаимодействия по тем или иным причинам нежелательно, в системе предусмотрена возможность «ручной» разработки соответствующих элементов.

Крупные КИС разрабатываются чаще всего в соответствии с каскадной (водопадной) или итеративной методологиями проектирования. Radix содержит в себе все необходимые средства для поддержки всех стадий жизненного цикла системы, включающего в себя:

проектирование модели; кодирование; помодульную и комплексную отладку; тестирование; выпуск релизов, дистрибутивов и пакетов обновлений для клиентов; установку, тестирование и ввод в эксплуатацию обновлений; исправление выявленных проблем; изготовление патчей и т. д. Более того, актуальная стадия жизненного цикла проекта является одной из характеристик проекта, и, в соответствии с движением проекта по стадиям жизненного цикла, разработчикам становится доступной та функциональность платформы, которая необходима именно на данной стадии. Сказанное, впрочем, не означает, что с помощью Radix невозможно разрабатывать проекты с использованием других методологий (например, Agile). Более того, уникальный подход, предусматривающий автоматическую генерацию диалогового интерфейса по требованиям, зафиксированным в модели, позволяет создавать прототипы будущей системы за минимальное время, что дает возможность максимально быстро и качественно уточнять требования и добиваться внушительного прогресса даже при самых коротких итерациях, практикуемых в Agile.

Особое внимание в Radix уделено повышению эффективности разработки проектов. Многолетний опыт работы на рынке сложных программных систем привел разработчиков Radix к модели бизнеса, в чем-то схожей с моделью, используемой разработчиками платформы «1С», – выпуск тиражируемого программного продукта, который может быть затем сконфигурирован под потребности конкретного заказчика. Конфигурирование системы, включающее изменение поведения, набора и структуры хранимых данных, не обязательно должно выполняться самим разработчиком: в этой роли может выступать либо сам заказчик, либо сторонние организации. Такой подход позволяет минимизировать затраты и разработчика, и заказчика на сопровождение системы. Radix содержит все необходимые для этого средства и обеспечивает поддержку как параметрического (включение/отключение той или иной опции и настройка ее параметров), так и алгоритмического (самостоятельное написание нового способа обработки данных, создание нового отчета, подключение к новым сервисам и т. д.) конфигурирования, что позволяет каждому пользователю настроить приобретенную систему в соответствии со своими потребностями, не обращая при этом к разработчику, не дожидаясь очередной версии и даже без остановки работающей системы.

Radix как платформа разработки не был бы достаточно эффективен, если бы не содержал в себе набор готовых технологических компонентов, в числе которых:

– компоненты SOA: системные сервисы, каталог сервисов,

маршрутизация, безопасность и т. д.;

– Workflow Engine для оркестровки бизнес-процессов (в том числе с человеческим участием);

– система разграничения доступа;

– средства аудита действий пользователей;

– средства аудита изменения хранимых данных;

– поддержка различных каналов персональных коммуникаций (Email, SMS, Push-оповещения и т. д.);

– планировщик заданий;

– дизайнер и генератор отчетов;

– средства описания и поддержки различного рода протоколов (включая, но не ограничиваясь протоколами на основе XML);

– средства генерации и контроля лицензий на право использования программного продукта и отдельных модулей.

Использование этих компонентов позволяет решить ряд типовых задач, имеющих место в различных КИС. Набор составных частей платформы постоянно расширяется.

В целом, можно утверждать, что платформа RadixWare является стабильным и эффективным средством разработки проектов в области КИС, позволяющим минимизировать затраты разработчика на создание и сопровождение проекта. Вместе с тем Radix – динамично развивающийся проект, открытый для новых идей. Одной из причин, по которой разработчики Radix перевели проект в разряд общедоступных, являлось желание привлечь сторонних разработчиков и энтузиастов программирования к развитию нового, интересного и полезного проекта.

Автоматизация тестирования с помощью Selenium

Сорокин М.О. к.т.н.,

*Авиационный учебный центр Объединенной федерации
сверхлегкой авиации, г. Москва*

Аннотация. В статье приводится описание набора инструментов автоматизированного тестирования программных средств. Приводится пример использования Selenium на учебном проекте. Сделан вывод о целесообразности применения пакета программ для снижения временных расходов на разработку тестов.

Ключевые слова: тестирование, автоматизация, Selenium.

Тестирование программного обеспечения — неотъемлемая составляющая процесса разработки программного обеспечения (ПО). Проследив развитие моделей процессов разработки легко обнаружить, что со временем тестирование становится не отдельной активностью, характеризующей определенный этап разработки, а наоборот — становится неотъемлемой частью процесса.

По степени автоматизации различают ручное тестирование и автоматизированное. При ручном тестировании, тест-кейсы выполняются человеком вручную. Автоматизированное тестирование напротив, требует участие человека только для создания и поддержания актуальности самого набора тест-кейсов. Выполнение тестов и проверка результатов осуществляется соответствующим ПО. В целом, автоматизация способствует:

- частому выполнению регрессионного тестирования;
- снижению неопределенности о текущем качестве программного обеспечения;
- способствует проведению стресс тестов и может являться основой для проведения нагрузочного тестирования;
- реализации принципов гибкой разработки (автоматизированное тестирование становится частью процесса непрерывной интеграции).

Одним из инструментов, позволяющих автоматизировать тестирование веб-приложений, является Selenium. История проекта Selenium уходит в 2004 год. Изначально проект представлял собой библиотеку на JavaScript, разработанную Джейсом Хаггинсом (Jason Huggins) при работе над внутренними проектами в компании ThoughtWorks. Со временем, Selenium вырос в набор программных инструментов, которые будут рассмотрены ниже.

SeleniumRC (Selenium 1) — результат развития Selenium, уходящий корнями в 2004 год. Представляет собой механизм для запуска тест-кейсов в браузерах. Представлял собой реализацию на JavaScript. В настоящее время является устаревшей версией.

SeleniumWebDriver (Selenium 2) — актуальная ветвь развития проекта. Рекомендуется для разработки новых тестов. В настоящее время поддерживает множество браузеров и операционных систем, в том числе Safari и встроенные браузеры операционных систем Android и iOS. В отличие от SeleniumRC для управления браузером используются собственные методы, предоставляемые браузерами (у различных браузеров они разные). Ключевым является стандартизированное API, позволяющее управлять действиями на странице. Поддерживается поиск

элементов, получение текстовых значений, заполнение форм, навигация, работа с Cookies& операции перетаскивания.

SeleniumIDE - интегрированная среда разработки тестов. Технически представляет собой дополнение для браузера Firefox. Позволяет записать действия пользователя на странице а затем воспроизвести этот сценарий. Существует возможность экспорта записанного сценария в виде программного кода на многих языках программирования. Содержит функции для проверки информации на странице. Такие проверки и являются критерием успешного прохождения теста. Среда позволяет организовать набор тест-кейсов и запускать их последовательно. Разработчиками позиционируется как средство прототипирования, поэтому наличие функций итерирования или условных операций не предполагается. Вполне логичным представляется разработка базового сценария тестирования в среде SeleniumIDE, экспорт сценария в виде программного кода и реализация необходимых операций.

Selenium-GRID – инструмент, позволяющий масштабировать выполнение набора тестов на несколько сред исполнения тестов. С одной стороны — это может ускорить выполнение всего набора тестов, так как разные тест-кейсы будут запущены из разных средах параллельно. С другой — возможны ситуации, при которых необходима проверка корректности работы системы при параллельном выполнении одного тест-кейса.

Selenium-Server – инструмент, позволяющий подключиться к удаленной машине, на которой имеется версия браузера, в котором необходимо запустить тесты. Кроме того Selenium-Server является компонентом, необходимым для реализации распределенного выполнения тестов с помощью Selenium-GRID.

Несмотря на то, что возможности SeleniumIDE не предполагают разработку сложных текст-кейсов, содержащих циклы и условные операторы, применение в проекте хотя бы этого инструмента может значительно повысить производительность труда команды. Например, такие сценарии можно использовать в системе учета дефектов проекта — где вместо описания шагов по воспроизведению дефекта в приложении тестировщик может приложить готовый сценарий. Таким образом, разработчику останется только скачать сценарий, воспроизвести его и однозначно воспроизвести ситуацию. Дополнительно, с помощью сторонних разработок можно реализовать запуск тест кейсов в формате html – выходном формате сохранения сценариев SeleniumIDE. Одним из таких средств является проект SeleneseRunner.

Проект **SeleneseRunnerJava**, размещенный на сервисе GitHub предоставляет возможности интерпретации скриптов SeleniumIDE. Инструмент представляет собой java приложение, которое позволяет выполнить тест-кейс в браузере, используя при этом заданный драйвер браузера. Технически приложение на основании сценария тестирования выполняет вызовы SeleniumWebDriver, таким образом выполняя сценарий в браузере. Примечательно, что приложение является гибко настраиваемым, а все основные параметры можно задать в качестве параметров командной строки. Используя это средство легко реализовать принцип непрерывной интеграции при разработке программного обеспечения. Для этого лишь необходимо разрабатываемые сценарии в SeleniumIDE помещать в отдельный репозиторий, обновления в котором проверяются сервером автосборки (например Jenkins). При обновлении репозитория Jenkins может запускать тесты на выполнение с помощью инструмента Selenese-Runner. Примечательно, что результат выполнения доступен как в формате xml, так и в формате html страницы, а приложение после окончания работы выдает код возврата, по которому можно определить успешность выполнения тестов.

Рассмотрим применение средств SeleniumIDE и Selenese-Runner на примере простого проекта. Предположим, что имеем тестовый проект с реализацией простого калькулятора. Программа подразумевает две страницы: страница авторизации и непосредственно с калькулятором. На странице авторизации пользователь вводит логин и пароль и после нажатия на кнопку «Вход» - попадает в систему. Внешний вид форм представлен на рисунках 1 и 2.

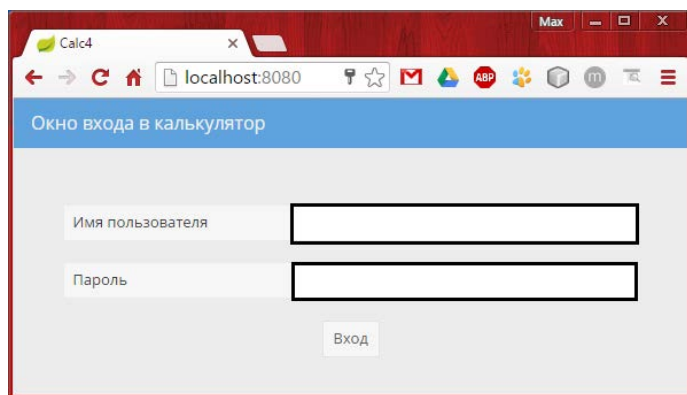


Рис. 1. Страница авторизации

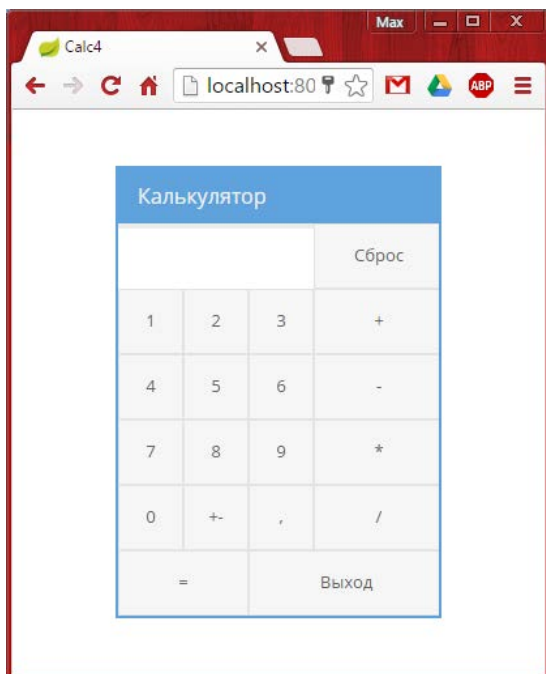


Рис 15. Страница калькулятора

Для примера, сформулируем три тестовых кейса:

1. Авторизация в приложении и успешный вход на страницу калькулятора.
2. Ввод цифр на цифровой панели калькулятора и проверка корректности отображения введенных символов в текстовом поле отображения чисел и результатов вычисления.
3. Проверка операции сложения. Для этого осуществляется вход в приложение, ввод одного числа, затем нажатие на кнопку операции сложения, ввод второго числа. Затем выполняется нажатие на кнопку получения результата и проверяется корректность результата вычисления.

Для создания тест-кейсов воспользуемся утилитой SeleniumIDE. Рабочее окно представлено на рисунке 3. Для создания кейса понадобятся следующие элементы управления: кнопка старта и остановка записи тестового кейса (1), кнопка запуска тест кейса на исполнение (2), кнопка запуска набора тестов (3). Особое внимание следует уделить блоку элементов (4). Эти элементы предназначены для

создания и редактирования этапов выполнения тест-кейса. Стоит отметить большой выбор команд: в простейшем случае это может быть эмуляция нажатия кнопок мыши на элементах, в более сложных — доступна возможность запрограммировать код этапа выполнения на языке JavaScript.

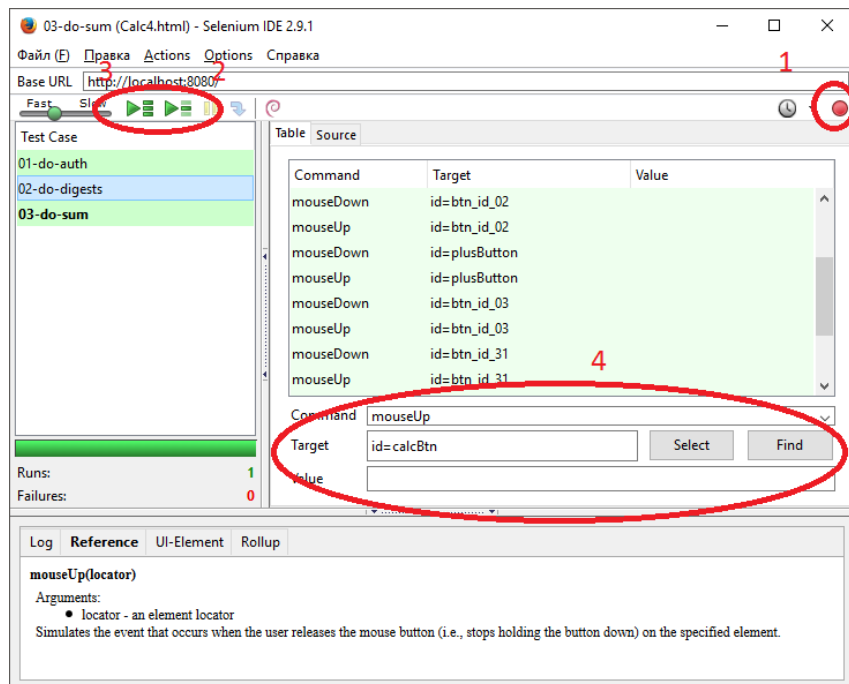


Рис.16. Внешний вид утилиты SeleniumIDE

После создания тест-кейсов и их успешного выполнения в среде SeleniumIDE, рассмотрим запуск созданного набора тестов с помощью Selense-Runner. Для этого воспользуемся следующей командой:

```
java -jar selense-runner.jar Calc4.html --set-speed 3000 --html-result
```

Здесь Calc4.html – сохраненный файл набора тестов, --html-result . - указание каталога, в который поместится отчет о результате выполнения тестов, в формате html (в примере используется текущая директория). По умолчанию, утилита использует драйвер для браузера Firefox, поэтому результатом ввода команды будет запуск окна браузера, выполнение тестов и формирование отчета. Пример отчета по результатам выполнения тестов представлен на рисунке 4.

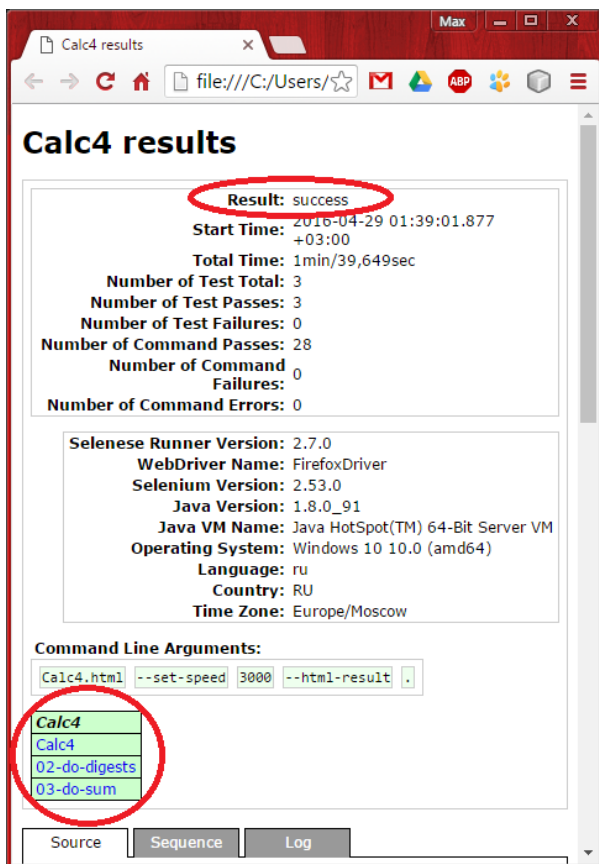


Рис. 17. Отчет с результатами выполнения набора тестов

Применяя эти инструменты в собственных проектах, можно значительно повысить эффективность работы: рутинные операции автоматизируются, а продуктовая команда всегда имеет актуальную информацию о состоянии продукта в любой момент времени. Разумеется, сопровождение автоматизированных тестов и их актуализация является «проектом в проекте»: развиваться тесты должны параллельно с развитием продукта. Однако, затраты на старт хотя бы с применением SeleniumIDE – минимальны и, в то же время, оправдывают себя за счет значительной экономии времени на проведение тестирования. Кроме того, наборы тестов, сформированные в SeleniumIDE, со временем можно выгрузить в качестве программного кода, и уже на его основе при необходимости получить ещё большую гибкость. В рыночных условиях

немаловажен факт бесплатности инструментария Selenium – продукт распространяется по лицензии Apache 2.0.

Список используемых источников

1. Selenium - WebDriverAutomotion. – Режимдоступа: <http://www.seleniumhq.org/>
2. GitHub – vmi/selenese-runner-java: SeleniumIDE native format interpreter. – Режимдоступа: <https://github.com/vmi/selenese-runner-java>
3. Куликов С. Тестирование программного обеспечения. Базовый курс. – Режимдоступа: http://svyatoslav.biz/software_testing_book/

Преимущества и недостатки объектно-ориентированного моделирования на основе открытого стандарта UML

*Багатурия В. В., Сусов Р. В., к.э.н.,
ФГБОУ ВО «Московский Государственный Университет Леса»*

Аннотация. В статье приводится описание открытого языка моделирования и проектирования программного обеспечения и информационных систем для автоматизации бизнес-процессов UML. Во многом благодаря тому, что UML - это открытый стандарт и существует множество свободно распространяемых программных продуктов, поддерживающих нотации этого стандарта, UML успешно применяется на практике в различных областях, таких как разработка программного обеспечения, образование, бизнес.

Ключевые слова: UML, ограничения, недостатки

Введение. UML (Unified Modeling Language) — совокупность графических нотаций для разработки объектно-ориентированных моделей программного обеспечения, организационных структур, бизнес-процессов. Используемые в UML модели опираются на теорию систем, цель которой состоит в описании сложных систем при помощи единообразных стандартов. Системы состоят из множества компонентов, связанных между собой различными отношениями. Исходя из принципа выделения в теории систем структуры и поведения системы, в UML включены объектно-ориентированные модели, предназначенные для описания структуры системы и моделирования поведения системы соответственно. При моделировании структуры системы основная цель состоит в формировании классов и их связей с действиями, динамическое поведение системы описывается через обмен сообщениями [1].

Объектно-ориентированное моделирование состоит в описании значимых сущностей реального мира с целью понимания их взаимодействия [2], что достигается путем создания моделей системы в виде набора классов и объектов предметной области. При этом система

представляется с использованием иерархии классов, а ее функционирование рассматривается как взаимодействие совокупности объектов [3]. Объектно-ориентированная разработка программного обеспечения основывается на концепции первоочередной оптимизации автоматизируемых бизнес-процессов или их частей, которые выступают основной для разработки программного обеспечения в виде сценариев работы [1].

Изначально UML был создан для проектирования программного обеспечения, поэтому часто ассоциируется именно с моделированием объектно-ориентированных программных систем. Однако в настоящее время область использования UML не ограничивается моделированием программного обеспечения. UML имеет намного более широкое применение благодаря наличию разнообразных нотаций, позволяющих представить систему с различных сторон и свойственной UML расширяемости [4]. Например, нередко его используют для моделирования бизнес-процессов и отображения организационных структур. В настоящее время UML предоставляет коллекцию лучших инженерных практик, которые дают проверенные результаты в моделировании больших и сложных систем, поэтому UML претендует на роль открытого стандарта в области объектно-ориентированного анализа и проектирования информационных систем для поддержки бизнес-процессов [5]. Кроме этого, UML все более становится похож на стандарт моделирования бизнес-процессов и рассматривается как фундамент для их моделирования, но не только потому, что он становится стандартом моделирования, но также потому, что он хорошо применим на практике.

Моделирование с использованием UML поддерживается различными как коммерческими, так и свободно распространяемыми инструментальными средствами. К наиболее популярным коммерческим продуктам можно отнести RationalRose, ARISToolset, Microsoft Visio, UMLStudio, среди свободного программного обеспечения можно выделить UMLet, yEd, DiagramDesigner. Некоторые инструментальные средства, основанные на UML, представляют собой CASE-технологии, которые позволяют также автоматизировать этапы анализа и проектирования программного обеспечения, а также генерировать код на различных языках программирования (например, C++) и выпускать проектную документацию. Кроме того, в подобные инструментальные средства могут быть включены функции реинжиниринга программ, обеспечивающие повторное использование программных компонент в новых проектах.

Благодаря тому, что с одной стороны UML - это открытый стандарт, а с другой стороны, существует множество свободно распространяемых программных продуктов, поддерживающих нотации этого стандарта, UML получил широкое распространение и используется

в различных сферах деятельности: в разработке программного обеспечения, образовании, бизнесе.

Диаграммы UML. UML позволяет создавать несколько типов визуальных диаграмм:

- диаграмма действий — activity diagram;
- диаграмма класса — class diagram;
- диаграмма описания класса — class description diagram;
- диаграмма взаимодействия — collaboration diagram;
- диаграмма компонент — component diagram;
- диаграмма состояний — state chart diagram;
- диаграмма вариантов использования — use case diagram.

Диаграммы действий показывают функциональные потоки в системе и могут использоваться в моделировании бизнес-процессов для демонстрации рабочих потоков в организации. На этих диаграммах определяются начальные и конечные точки потоков, действия в рамках этих рабочих потоков, а также последовательность этих действий [6]. В языке UML действия всегда относятся к объектам. Таким образом, диаграммы действий привязываются к классу, операции или использованию приложений и описывают относящийся к ним внутренний процесс [7].

Диаграмма класса - это статическая структурная диаграмма, описывающая структуру системы. Диаграмма класса отражает взаимодействие между классами системы. Классы содержат данные и поведение (действия), влияющее на эти данные [6]. С концептуальной точки зрения диаграммы классов описывают понятия изучаемой предметной области и увязывают эти понятия в строгую систему. Каждый класс можно трактовать как некоторое сложное понятие моделируемой предметной области. Диаграмма классов определяет типы объектов системы и различного рода статические связи между ними. В качестве объектов предметной области могут рассматриваться конкретные предметы или реальные сущности (например, клиент или заказ).

Диаграммы описания классов предназначены для более полного и точного графического представления связей между элементами. Диаграмма описания класса — это дополнение к стандартным диаграммам UML, которое позволяет более точно определить классы. Параметры диаграммы описания класса являются подмножеством параметров диаграммы класса. Другими словами, все параметры моделирования в диаграмме описания класса доступны и в диаграмме класса [7].

Диаграмма взаимодействия отражает взаимодействия в форме обмена сообщениями между объектами. Объекты, называемые также экземплярами, — это конкретные экземпляры классов [7].

Диаграмма компонент - статическая структурная диаграмма, показывающая разбиение программной системы на структурные компоненты и связи (зависимости) между компонентами. Компоненты — это элементы, которые формируют блоки в процессе компиляции или компоновки, а также во время выполнения системных операций [7].

Диаграмма состояний предназначена для моделирования различных состояний, в которых может находиться объект. В то время, как диаграмма классов показывает статическую картину классов и их связей, диаграммы состояний применяются при описании динамики поведения системы [6].

Диаграмма вариантов использования отображают взаимодействие между вариантами использования, представляющими функции системы, и действующими лицами, представляющими людей или системы, получающие или передающие информацию в данную систему. Диаграмма вариантов использования описывает внешнее поведение системы с точки зрения пользователей. Диаграмма вариантов использования является основной для общения и взаимопонимания между пользователями, заказчиками и разработчиками [8]. Основное предназначение диаграммы вариантов использования - предоставить единое средство, дающее возможность заказчику, конечному пользователю и разработчику совместно обсуждать функциональность и поведение системы.

Преимущества и недостатки UML. Значительное достоинство диаграмм действий UML признается в том факте, что они поддерживают моделирование параллельного поведения системы, что делает UML мощным инструментом для моделирования потоков работ организации. Благодаря предоставлению возможности для моделирования как параллельного, так и последовательного поведения, диаграммы действий UML - хороший инструмент для разработки согласованной модели, которая впоследствии будет использоваться для имитационного моделирования. Это также важная причина для выбора этих диаграмм при подготовке моделей к имитационному моделированию – они могут управлять параллельными процессами в отличие от большинства альтернативных нотаций. Многие авторы изучают пригодность и соответствие диаграмм действий UML для построения полезных процессных и имитационных моделей [9].

Еще одним неоспоримым достоинством объектно-ориентированного моделирования является тесная связь моделей с реализацией. Это предельно облегчает, например, создание прототипов систем. Однако, поскольку объектно-ориентированное моделирование

строго придерживается концепции системной разработки, основное внимание при таком подходе сосредоточено не на вопросах бизнеса. Архитектура ARIS, например, напротив, изначально сфокусирована именно на бизнес-процессах [1]. Иначе говоря, моделирование с помощью UML не позволяет уделять достаточное внимание вопросам бизнеса.

В контексте применения диаграмм в качестве спецификации бизнес-процессов выделяются сильные и слабые стороны диаграмм действий UML. Сильные стороны диаграмм действий относительно альтернативных методик, предлагаемых коммерческими системами проектирования бизнес-процессов, по существу следующие:

- UML поддерживают отправление и получение сигнала на концептуальном уровне;
- поддерживают оба состояния ожидания и состояния обработки данных;
- предоставляют однородный механизм для декомпозиции действий на поддействия.

Комбинация возможности декомпозиций и отправления сигнала представляет собой убедительный подход к обработке прерывания действий. К недостаткам диаграмм действий относятся: некоторые из конструкций диаграмм действий испытывают недостаток в точном синтаксисе и семантике; они не полностью поддерживают важные виды синхронизации [10].

При моделировании системы задача заключается в том, чтобы упростить рассматриваемый объект путем абстрагирования. Несмотря на абстрагирование от неактуальных свойств рассматриваемого объекта, в объектно-ориентированном моделировании с использованием UML сохраняется значительный объем семантики, в частности, при определении связей класса с атрибутами, методами и ассоциативными отношениями. Семантическое описание призвано сделать модель интуитивно понятной. С другой стороны, избыток семантики ведет к чрезмерному усложнению больших моделей, затрудняя их понимание даже для квалифицированных пользователей [3].

Еще одним недостатком объектно-ориентированного подхода является невозможность достаточно детального описания процессов. Даже используя такие диаграммы, как диаграмма вариантов использования или диаграмма взаимодействий, трудно наглядно представить ветвление процессов, организационные аспекты и потоки выходов [3]. С целью связи процессно-ориентированных моделей, разрабатываемых для повышения эффективности деятельности организации, и объектно-ориентированных моделей, используемых при разработке информационных систем, поддерживающих выполнение

бизнес-процессов, разработана методика интеграции UML и EPC диаграмм [11]. Методика включает описание эквивалентных элементов и зависимостей между EPC и UML диаграммами. Для полной интеграции объектно-ориентированного и процессно-ориентированного подходов в объединенную теорию бизнес-моделирования рассматриваются четыре принципа: класс (атрибуты, методы, жизненный цикл), наследование, взаимодействие и масштабирование [11].

Самым серьезным недостатком объектно-ориентированного подхода UML является невозможность достаточно детального описания и оптимизации бизнес-процессов. Кроме этого, избыток семантики UML ведет к чрезмерному усложнению моделей, затрудняя их понимание заинтересованными лицами. Несмотря на это многими авторами исследуется адекватность и полнота диаграмм действий UML для применения их в качестве стандарта в области процессного моделирования путем систематической оценки возможностей поддержки ими моделирования всей совокупности шаблонов потоков работ. Исследования показывают, что диаграммы действий способны поддерживать возникающие на практике ситуации, которые не могут быть отработаны большинством коммерческих систем управления потоками работ. С другой стороны, диаграммы действий не в состоянии отработать некоторую часть практических ситуаций, что указывает на направления для их совершенствования [10].

Заключение

UML хорошо применим в области объектно-ориентированного анализа и проектирования программного обеспечения и информационных систем для автоматизации бизнес-процессов. Важным достоинством UML является тесная связь моделей с их реализацией в информационных системах. Это значительно облегчает создание прототипов информационных систем для поддержки бизнес-процессов. Во многом благодаря тому, что UML - это открытый стандарт и существует множество свободно распространяемых программных продуктов, поддерживающих нотации этого стандарта, UML успешно применяется на практике в различных областях, таких как разработка программного обеспечения, образование, бизнес.

Список использованных источников

1. Шеер А.-В. Моделирование бизнес-процессов — М.: Изд. АОЗТ “Просветитель”, 2000. — 205 с.
2. Шлеер С. Объектно-ориентированный анализ: Моделирование мира в состояниях / С. Шлеер, С. Меллор. — Киев: Диалектика, 1993. — 240 с.

3. Войнов И.В. Моделирование экономических систем и процессов. Опыт построения ARIS-моделей / И.В. Войнов, С.Г. Пудовкина, А.И. Телегин — Челябинск: Изд. ЮУрГУ, 2002. — 395 с.
4. Арлоу Д. UML 2 и Унифицированный процесс. Практический объектно-ориентированный анализ и проектирование, 2-е издание / Д. Арлоу, И. Нейштадт; Пер. с англ. — СПб: Символ-Плюс, 2007. — 624 с., ил.
5. Вендров А.М. CASE-технологии. Современные методы и средства проектирования информационных систем. М.: Финансы статистика, 1998. — 176 с.
6. Богс У. UML и Rational Rose 2002 / У. Богс, М. Богс — Изд. “ЛОРИ”, 2004. — 509 с.
7. Каменнова М.С. Моделирование бизнеса. Методология ARIS. / М.С. Каменнова, А.И. Громов, М.М. Ферапонтов, А.Е. Шматалюк, 2001. — 333 с.
8. Фаулер М. UML. Основы. / М. Фаулер, К. Скотт ; Пер. с англ. — СПб: Символ-Плюс, 2002. — 192 с., ил.
9. Barjis J., Shishkov B. UML based business systems modeling and simulation. In Proceedings of 4th International EUROSIM Congress “Shaping future with simulation”. Aula Conference Centre Delft, The Netherlands, June 26-29, 2001.
10. Marlon Dumas, Arthur H.M. ter Hofstede. UML Activity Diagrams as a Workflow Specification Language. In Proceedings of 4th International Conference on the Unified Modeling Language "Modeling Languages, Concepts and Tools", Toronto, Ontario, Canada, October 1 [Electronic resource] — 5, 2001. — Mode of access: http://math.ut.ee/~dumas/uml01_dumas.pdf
11. Desel J., Oberweis A., Reisig W., Rozenberg G. Petri Nets and Business Process Management. [Electronic resource] — Mode of access: <http://www.dagstuhl.de/Reports/98271.pdf>

Патентный поиск визуализации теоретико-множественного анализа сложных систем

*Чеканова Е. Д.,
ФГБОУ ВО «Магнитогорский государственный технический
университет им. Носова»*

Аннотация. В данной статье приведены результаты патентного поиска по теме «Визуализация результатов теоретико-множественного анализа сложных систем». По итогам проанализированы два патента, зарегистрированные в федеральной службе по интеллектуальной собственности Российской Федерации. Проведен поиск заявок на

патенты и поиск программ для ЭВМ. Выделены три программы и проведен их анализ. Представлена информация об их специализациях.

Ключевые слова: моделирование, визуализация, анализ систем.

Введение

В настоящее время нас окружает множество различных систем. Их количество растет и для выполнения все более и более растущих потребностей человека они должны модернизироваться и совершенствоваться. Для этого необходимо такие системы изучать. Системный анализ и теория систем позволяют изучать закономерности построения, функционирования, развития систем и методы их исследования, в том числе используя для этого ЭВМ [1-11].

Проведение теоретико-множественного анализа позволяет выявлять структуру системы. В зависимости от выдвинутых целей и изучаемой системы в рамках теоретико-множественного анализа в системе выделяются элементы системы, их свойства и взаимодействия друг с другом. Таким образом, система представляется как множество подсистем. Результатом теоретико-множественного анализа является графическое представление системы в виде блок-схемы, в которой элементы обозначаются определенными фигурами блок-схемы, а взаимосвязи между ними – соединительными линиями. Кроме того, необходимо описание системы, для этого используются три таблицы: таблица, описывающая элементы системы, таблица – описание взаимосвязей между элементами и таблица для описания свойств элементов системы.

На данный момент перед исследователем стоит большой выбор различных программных продуктов, позволяющих оформить результаты проведенного теоретико-множественного анализа. В том числе, программные продукты компании Microsoft, такие как Word, Excel, Visio. Они предоставляют большой выбор инструментов для создания схем, записей и таблиц по отдельности, однако работа с несколькими программами отнимает много времени. Помимо этого, исследуемая система может содержать большое количество элементов, и каждый элемент может рассматриваться как отдельная система. Следует отметить, что в таком случае, безошибочное создание двух и более отдельных файлов, описывающих систему затруднительно, что является проблемой.

Проведенный анализ информации показал актуальность выполнения теоретико-множественного анализа и выявил проблемы в данной области [12-14]. Учитывая это, была поставлена следующая цель исследования: сокращение времени, необходимого на проведение и оформление результатов теоретико-множественного анализа сложной системы.

Патентный поиск изобретений

Решено провести патентный поиск для изучения изобретений и выданных авторских прав среди патентов, зарегистрированных в Российской Федерации [15]. В качестве предмета поиска были выбраны следующие формулировки: моделирование и визуализация систем. Результаты поиска представлены в таблице 1.

Найдено два зарегистрированных патента. Представленные патенты отражают решения поставленных задач, которые необходимо решить для достижения цели исследования. Рассмотрим их подробнее.

Патент №2294015 «Способ автоматического моделирования системы управления процессом и система управления процессом» рассматривает способ автоматического моделирования системы управления процессом за счет организации пользовательского интерфейса в виде древовидной структуры, отражающей топографию элементов в системе. Построение данных в виде дерева позволяет структурировать их и быстро находить необходимый элемент по идентификатору, на основе которого построено дерево. Однако поиск этого же элемента потребует больше времени и ресурсов, если использовать другой параметр элемента, не являющийся идентификатором. Более эффективный способ хранения такого рода информации это базы данных, в которые входили бы отдельно описание элементов и отдельно данные о взаимосвязях [16].

Патент №2285288 «Способ фиксации и визуализации вида изменяющегося объекта в любой из моментов или периодов времени (варианты)» рассматривает возможность расчета и сохранения данных о положении элемента на схеме, для этого предлагается выделение дополнительного блока памяти, для определения место положения элемента, а так же времени в течении которого данный элемент будет неподвижен. На данной стадии разработки программного продукта позволяющего визуализировать результаты теоретико-множественного анализа не предусматривается возможность произвольного движения элементов системы на схеме, однако предполагается, что исследователь будет иметь возможность редактировать положение элементов на схеме относительно друг друга. Из этого следует, что сохранение данных о положении элемента необходимо, а вот в данных о времени нахождения в покое нет необходимости. Для хранения данной информации все так же можно использовать базу данных. Однако нельзя допустить возможность редактирования координат элемента аналогично данным описывающих свойства элементов. Так, например, название элемента пользователь может придумать любое, а вот координаты положения элемента системы на схеме пользователю проще определить, ориентируясь на положение других элементов в системе, а не по данным их координат, т.е. данные о положении элемента, а именно координаты лучше спрятать [17].

Таблица 1.

Патентная документация (патенты на изобретения и полезные модели) отобранные для последующего анализа

Страна выдачи, вид и номер охранного документа, классификационный индекс МПК	Заявитель, страна. Номер заявки, дата подачи заявки, дата приоритета, конвенционный приоритет	Название изобретения, полезной модели.
RU, патент на изобретение №2294015, G06N7/06, G05B19/042	Фега грисхабер кг, Германия. Заяв. №2005109473/09, 01.04.2005. Приор. 02.04.2004 US 10/816,666. Оpubл. 10.10.2006	Способ автоматического моделирования системы управления процессом и система управления процессом
RU, патент на изобретение №2285288, G06F17/40, G06N7/06	Общество с ограниченной ответственностью "Научно-производственное предприятие "Гарант-Сервис", Россия. Заяв. №2004124354/09, 11.08.2004. Приор. 11.08.2004. Оpubл. 10.10.2006	Способ фиксации и визуализации вида изменяющегося объекта в любой из моментов или периодов времени (варианты)

Патентный поиск заявок на изобретения

Так же необходимо проверить и уже поданные, но еще не проверенные патенты на изобретения – заявки. Поиск проводился по заявкам, поданным в реестре изобретений Российской Федерации [15]. В качестве предмета поиска были выбраны формулировки: моделирование и визуализация систем. Однако поиски не увенчались успехом. Таким образом, на данный момент нет заявок на получение патента по данной теме.

Поиск зарегистрированных программ для ЭВМ

Немаловажным является и поиск зарегистрированных программ для ЭВМ, позволяющих решать проблему по визуализации результатов теоретико-множественного анализа.

Во время поиска использовались те же ключевые слова, что и при проведении двух предыдущих. В результате были выбраны следующие три программы.

«Программа для анализа динамических свойств сложных систем» (свидетельство о государственной регистрации №2016610323) позволяет проводить анализ характера поведения сложных динамических систем, так же применима при проектировании различных устройств

электротехники, при моделировании и разработке импульсных источников питания, релейных систем автоматического управления и систем передачи информации. Так же позволяет определять и визуализировать характер траекторий движения переменных системы и характер движения изображающей точки системы на фазовой плоскости [18].

Программа для ЭВМ, имеющая свидетельство о государственной регистрации №2013613236 – «Система расчета и моделирования электрических кабельных трасс и трубопроводных систем для сложных изделий, характерных для высокотехнологичных отраслей машиностроения (ТРАССА)». Данная программа предназначена для проектирования сложных систем кабельных и трубопроводных трасс сложных изделий, характерных для высокотехнологичных отраслей машиностроения. Однако интересны её следующие функции:

- возможность формирования, хранения и визуализации геометрической информации, специфичной для реализуемых задач; возможности ввода и вывода информации об объекте проектирования с целью использования этой информации в процессе проектирования или передачи в другие системы САПР;

- возможность ручного и полуавтоматического анализа кабельных трасс и трубопроводов на предмет выполнения геометрических ограничений; возможность формирования, редактирования, хранения и вывода конструкторской и сметной документации на проектируемую систему.

Данная программа предназначена для сокращения времени расчета и моделирования электрических кабельных трасс и трубопроводных систем сложных изделий [19].

«Программа визуализации сетевых операций с использованием двудольных ориентированных графов – сетей Петри, PETR1», имеющая свидетельство о государственной регистрации №2014613285 предназначена для графического моделирования типовых сетевых операций с использованием полученных графических моделей для анализа технических систем за счет с использования методов анализа сетей Петри – аппарата теории графов. Программа позволяет в режиме реального времени разрабатывать гибкие функциональные модели сетевых операций с возможностью проведения анализа этих моделей в динамике. Результаты работы программы обеспечивают разработку графических диаграмм, моделирующих типовые сетевые операции. Имитационные модели позволяют описывать достаточно сложные объекты управления, используемые в авиационной и космической технике [20].

Анализ результатов поисков

В результате проведения патентного поиска изобретений было найдено два изобретения зарегистрированных в федеральной службе по интеллектуальной собственности.

В результате проведения патентного поиска были найдены два патентных документа интеллектуальной собственности, зарегистрированные в федеральной службе по интеллектуальной собственности Российской Федерации. Одно из них зарегистрировано нашим соотечественником Д.В. Першеевым, а второе – Ф. Файсту и А. Изенманну, подданными Германии. Каждое изобретение было подробно рассмотрено, выделены их преимущества и недостатки. Так же проведен поиск по заявкам на патенты, однако данный поиск не принес результатов.

Помимо этого проведен поиск программ для ЭВМ, выделены три, наиболее близкие к рассматриваемой теме программы, и проведен их анализ. Представлена информация об их специализациях.

Список использованных источников

1. Ильина Е.А., Чеканова Е.Д., Дёрина Н.В. О проблемах визуализации результатов теоретико-множественного анализа // Мир науки и инноваций. – 2015. – Т.3. – №2(2). – С.6-8.

2. Ilina E.A., Chekanova E.D., Dyorina N.V. The issues of set-theoretic analysis result visualization // SWorldJournal. – 2015. – T.J21510.– №2.– С.95-98.

3. Логунова О.С. и др. Визуализация результатов научной деятельности: учеб.пособие // Магнитогорск: Изд-во Магнитогорск.гос. техн. ун-та им Г. И. Носова, 2015. – 85 с.

4. Логунова О.С., Ильина Е.А. Методика исследования предметной области на основе теоретико-множественного анализа // Математическое и программное обеспечение систем в промышленной и социальной сферах. – 2012. – №2. – С. 281-291.

5. Чернышов В. Н. Чернышов А. В. Теория систем и системный анализ: учебное пособие // Тамбов: Тамб. гос. техн. ун-та, 2008. – 96 с.

6. Логунова О.С., Ильина Е.А. Структуризация лексикографической информации при разработке программного обеспечения // Математическое и программное обеспечение систем в промышленной и социальной сферах. – 2014. – № 1 (4). – С. 87-91.

7. Logunova O.S., Ilina E.A., Arefeva D.Y., Dyorina N.V. Index analysis of academic staff publication activity control // Математическое и программное обеспечение в промышленной и социальной сферах. – 2015. – № 1. – С. 43-47.

8. Логунова О.С., Ильина Е.А. Структуризация лексикографической информации при разработке программного

обеспечения // Математическое и программное обеспечение систем в промышленной и социальной сферах. – 2014. – №1(4). – С. 87-91.

9. Нургалина Р.Г., Ильина Е.А. Принятие решений при измерении уровня рефлексии в системе дистанционного обучения // Математическое и программное обеспечение систем в промышленной и социальной сферах. – 2012. – №2. – С. 250-256.

10. Чеканова Е.Д., Ильина Е.А. О визуализации результатов теоретико-множественного анализа // Актуальные направления научных исследований XXI века: теория и практика. – 2015. – Т. 3. – № 8-1 (19-1). – С. 385-389. DOI: 10.12737/15455.

11. Чеканова Е.Д. Разработка программного обеспечения для визуализации результатов теоретико-множественного анализа сложных систем // Ab ovo ... (С самого начала ...). – 2015. – С. 38-43.

12. Логунова О.С., Леднов А.В., Королева В.В. Результаты анализа публикационной активности профессорско-преподавательского состава ФГБОУ ВПО «Магнитогорский государственный технический университет им. Г.И. Носова» // Вестник Магнитогорского государственного технического университета им. Г.И. Носова. – 2014. – №3 (47). – С.78-87.

13. Логунова О.С., Ильина Е.А., Окжос К.М. Система оценки качества статей научного журнала // Математическое и программное обеспечение систем в промышленной и социальной сферах. – 2015. – № 2 (7). – С.56 -57.

14. Логунова О.С., Ильина Е.А., Окжос К.М. Система поддержки принятия решения для оценки качества статей научного журнала // Фундаментальные исследования. – 2016. – № 2 (3). – С.492-497.

15. Федеральное государственное бюджетное учреждение «Федеральный институт промышленной собственности»: [Электронный ресурс]. Режим доступа: . http://www1.fips.ru/fips_serv1/fips_servlet – свободный, дата обращения 12.04.16.

16. Способ автоматического моделирования системы управления процессом и система управления процессом [Текст]: пат. 2294015 Российская Федерация, МПК G06N7/06, G05B19/042/ Ф.Файст, А.Изенманн; заявитель и патентообладатель Фега грисхабер КГ. – № 2005109473/09, , заявл. 01.04.2005; опубл. 20.02.2007, Бюл. № 5.

17. Способ фиксации и визуализации вида изменяющегося объекта в любой из моментов или периодов времени (варианты) [Текст]: пат. 2285288 Российская Федерация, МПК G06F17/40, G06N7/06/ Д.В.Першеев; заявитель и патентообладатель Общество с ограниченной ответственностью "Научно-производственное предприятие "Гарант-Сервис". – № 2004124354/09, заявл. 11.08.2004; опубл. 10.10.2006, Бюл. № 28.

18. Программа для анализа динамических свойств сложных систем [Текст]: пат. 2016610323 Российская Федерация/ Р.А.Набоков, В.Г. Рубанов; заявитель и патентообладатель ФГБОУ ВПО «Белгородский государственный технологический университет им. В.Г. Шухова» – №2015660555, заявл. 05.11.2015; опубл. 20.02.2016.

19. Система расчета и моделирования электрических кабельных трасс и трубопроводных систем для сложных изделий, характерных для высокотехнологичных отраслей машиностроения (ТРАССА) [Текст]: пат. 2013613236 Российская Федерация/ заявитель и патентообладатель: Российская Федерация, от имени которой выступает Министерство промышленности и торговли Российской Федерации – №2013611442, заявл. 15.02.2013; опубл. 20.06.2013.

20. Программа визуализации сетевых операций с использованием двудольных ориентированных графов – сетей Петри, PЕТR1 [Текст]: пат. 2014613285 Российская Федерация/ О.Н.Третьякова, А.А.Иванов; заявитель и патентообладатель ФГБОУ ВПО «Московский авиационный институт (национальный исследовательский университет)» – № 2014610394, заявл. 24.01.2014; опубл. 20.04.2014.

21. Ильина Е.А., Чеканова Е.Д. Теоретико-множественная модель взаимодействия компонентов системы // APRIORI. Серия: Естественные и технические науки. – 2015. – № 6. – С. 19.

РАЗДЕЛ 4. СВОБОДНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ В АДМИНИСТРИРОВАНИИ

Защита корпоративной почты от спама средствами Spam Assassin

*Борисов С.А., Наношкин А.Г., Сапегина В. С.,
ФГБОУ ВО «Магнитогорский государственный технический
университет им Г.И. Носова»,
Иванов И.А.,
филиал ООО «Свод интернешнл» в Краснодарском крае*

Аннотация. Статья посвящена описанию методов борьбы со спамом с помощью свободной программы фильтрации Spam Assassin. Приводится описание процесса установки и настройки программы.

Ключевые слова: спам, фильтрация, Linux, Spam Assassin

Большинство современных компаний используют Интернет в качестве профессиональной деловой среды с сильными коммуникационными связями, поэтому проблема борьбы со спамом на данный момент очень актуальна. Спам с внедренными вирусными программами может уничтожить важную информацию или парализовать работу компании на несколько часов, и это, несомненно, приводит к финансовым потерям и утрате доверия клиентов и партнеров [6]. Но и электронная почта без опасных вложений снижает производительность труда сотрудников и пропускную способность каналов связи, а также увеличивает общие ИТ-расходы на «обслуживание» электронных писем, не имеющих отношения к рабочей деятельности компании.

На первый взгляд, убытки незначительные для отдельного пользователя, но в масштабах всей индустрии и даже отдельной крупной компании выглядят впечатляюще. По разным оценкам, на спаме компании теряют от \$50 до \$200 в год в расчете на одного офисного сотрудника. В России ущерб от спама оценивается в \$200-\$250 миллионов, а во всем мире составляет \$10 миллиардов ежегодно [9]. По этой причине защита от спама выступает необходимой задачей обеспечения информационной безопасности и защиты информации, так как мировые убытки от него составляют несколько десятков миллиардов ежегодно [2].

Существуют два основных метода защиты почтового сервера от спама:

- защита от поступления спама на этапе получения почтовым сервером;

- «отделение спама» от остальной почты уже после получения почтовым сервером [10].

В рамках первого метода наиболее популярны такие способы как использование DNS Black List (DNSBL), Greylisting и других средства для фильтрации спама.

При выполнении анализа содержимого письма наиболее популярны такие методы, как проверка по различным алгоритмам, поиск особых ключевых слов рекламного характера или на основе теоремы Бейеса [10]. Данные методы позволяют отсеивать около 90% спама еще на этапе доставки в почтовый ящик. Уже доставленную почту можно разметить средствами анализа содержимого письма, например, используя программу Spam Assassin. Данный продукт позволяет на основе особых алгоритмов добавлять в заголовки письма соответствующие строки, а пользователь, на основе почтовых фильтров в почтовом клиенте, может отфильтровать почту в нужные папки почтовой программы.

Ниже перечислены наиболее известные и распространенные средства для фильтрации спама.

DSPAM – свободное программное обеспечение, представляющее собой статистический спам-фильтр [3].

DNSBL – списки хостов, хранимые с использованием системы архитектуры DNS, используются для борьбы со спамом [4].

DomainKeys – система e-mail аутентификации, разработанная для проверки DNS домена отправителя e-mail сообщения и достоверности сообщения [5].

Spam Assassin – средство для фильтрации спама на почтовом сервере, т.е. создания и настройки анти-спам шлюза; предназначена для работы на корпоративном почтовом сервере.

Фильтрация спама в Spam Assassin основана на взаимодействии ключевых компонентов – оценочного сервиса, транспортного агента и базы шаблонов писем [1].

SpamAssassin поддерживает Байесовскую фильтрацию, обработку DNSBL, Sender Policy Framework, DomainKeys, DKIM, Razor и другие методы распознавания спама.

Spam Assassin написан на языке Perl. Обычно он используется для фильтрации входящей почты одному или нескольким пользователям. Он может быть запущен как отдельное приложение, как часть другой программы, либо как клиент (spamc), который взаимодействует с демоном (spamd). Последний способ имеет более высокое быстродействие, но в некоторых условиях может создать угрозу для безопасности.

Spam Assassin поставляется с большим набором правил, которые определяют, какие письма являются спамом, а какие нет. Большинство правил основано на регулярных выражениях, которые сопоставляются с

телом или заголовком сообщения, но Spam Assassin также использует и другие методики. В документации Spam Assassin эти правила называются «tests».

Каждый тест имеет некоторую «стоимость». Если сообщение успешно проходит тест, эта «стоимость» добавляется к общему баллу. «Стоимость» может быть положительной или отрицательной, положительные значения называются «spam», отрицательные «ham». Сообщение проходит через все тесты, подсчитывается общий балл. Чем выше балл, тем больше вероятность, что сообщение является спамом.

У Spam Assassin есть настраиваемый порог, при превышении которого письмо будет классифицировано как спам. Обычно порог таков, что письмо должно подойти по нескольким критериям; срабатывание только одного теста недостаточно для превышения порога [1].

SpamAssassin один из наиболее удачных инструментов, сочетающий большое количество методов анализа.

Установка SpamAssassin

Данный продукт предназначен для таких операционных систем как FreeBSD, Ubuntu и Linux. SpamAssassin поддерживает гибкие возможности настройки, из которых следует выделить: создание новых правил проверки почты и интеграцию с почтовыми серверами и фильтрами (postfix, qmail, sendmail, procmail).

Для большинства пользователей подойдут настройки по умолчанию, поэтому рассмотрим конфигурирование других компонентов системы для взаимодействия со Spam Assassin на примере ОС Ubuntu. Установка программы происходит из окна терминала, который можно запустить, выбрав на панели пункт «Приложения->Стандартные->Терминал», либо запустив консоль с помощью комбинации клавиш Ctrl+Alt+F1.

Теперь подключим Spam Assassin к серверу Postfix, так как этот сервер является самым популярным MTA (Mail Transfer Agent), позволяющим эффективно передавать письма адресатам.

Команда установки Postfix: *apt-getinstallpostfix*.

При конфигурировании пакета необходимо выбрать профиль первоначальной настройки. Вне зависимости от того, для чего предполагается использовать Postfix, всегда следует выбрать Internet Site.

Архитектура Postfix выполнена в стиле UNIX: простые программы выполняют минимальный набор функций, но выполняют их быстро и надежно. Во время простоя почтовой системы ненужные демоны могут прекращать свою работу, высвобождая тем самым память, а при необходимости снова запускаются master-демоном [7].

В файл */etc/postfix/master.cf* после строки *smtp inet n-smtpd* добавим строчку:
-ocontent_filter=spamassassin

И в конец этого файла вставим:

```
spamassassin unix-n n- -pipe
```

```
user=virtual argv=/usr/bin/spamc -f -e /usr/sbin/sendmail -oi -f  
${sender} ${recipient}
```

Теперь необходимо перезапустить сервисы Postfix и SpamAssassin:

```
#service postfix restart
```

```
#service spamassassin restart
```

На этом установка заканчивается.

Одной из самых замечательных возможностей Spam Assassin является его способность к самообучению. Самообучаться Spam Assassin может как на пропущенном спаме, так и на письмах, которые были несправедливо помечены, как спам(ham).

Эту возможность использовать довольно легко. Необходимо создать 2 почтовых ящика, в один из которых почтовый клиент будет передавать пропущенный спам, а в другой -- ham. Предположим, что используется формат maildir, пропущенный спам хранится в папке .uncaught/, а ham - в папке .ham/. Тогда для самообучения достаточно выполнить всего лишь две команды

```
sa-learn --spam /home/rust/.maildir/.uncaught/cur
```

```
sa-learn --ham /home/rust/.maildir/.ham/cur [8]
```

Это один из возможных способов настройки, который позволяет фильтровать спам с помощью Spam Assassin и Postfix.

Существуют другие способы сделать то же самое, например, воспользоваться Amavis, который использует Spam Assassin или также можно использовать демон Spam Assassin (spamd). Amavisd – это программная оболочка, которая может вызывать любое количество фильтрующих программ по обнаружению спама, антивирусов и т.п [7].

Фильтр Spam Assassin может быть предельно гибко настроен для борьбы со спамом. Он предоставляет администратору возможность как глобально определять правила фильтрации всей почты, так и переложить часть настроек на пользователей, которые смогут подстраивать работу Spam Assassin под свои предпочтения. Использование Spam Assassin позволит упростить жизнь пользователям электронной почты, а при достаточно внимательной настройке – добиться весьма хороших результатов при минимуме ложных срабатываний.

Список использованных источников

1. SpamAssassin [Электронный ресурс] – Режим доступа: <https://ru.wikipedia.org>

2. Защита корпоративной почты и методы борьбы со спамом [Электронный ресурс] – Режим доступа: <http://www.mytec.ru>

3. DSPAM [Электронный ресурс] – Режим доступа: <https://ru.wikipedia.org>

4. DNSBL [Электронный ресурс] – Режим доступа: <https://ru.wikipedia.org>
5. DomainKeys [Электронный ресурс] – Режим доступа: <https://ru.wikipedia.org>
6. Защита корпоративной почты от СПАМа и вирусов. [Электронный ресурс] – Режим доступа: <http://mdex-nn.ru>
7. Фильтрация почты. [Электронный ресурс] – Режим доступа: http://help.ubuntu.ru/wiki/руководство_по_ubuntu_server/
8. Postfix [Электронный ресурс] – Режим доступа: http://help.ubuntu.ru/wiki/руководство_по_ubuntu_server/
9. Гаврилова И.В. Организация защиты персональных данных в образовательных учреждениях // Новые информационные технологии в образовании Материалы VII международной научно-практической конференции. Российский государственный профессионально-педагогический университет. Екатеринбург, 2014. С. 509-513.
10. Гаврилова И.В., Гаврилов Д.О. Настройка контент-фильтрации в образовательных учреждениях // Новые информационные технологии в образовании Материалы VI Международной научно-практической конференции. 2013. С. 317-319.
11. Чернова Е.В., Попова Е.В., Попова И.В., Зленко И.В. Обеспечение безопасности системы информационно-аналитической поддержки научных исследований// Программные продукты и системы. 2009. № 4. С. 14.

Опыт сопровождения веб-приложения Redmine для автоматизации учета заявок на ремонт и обслуживание офисной техники компании

*Коновалов В.Ю.,
ООО «Корпоративные системы Плюс»*

Аннотация: в статье рассматривается актуальность использования веб-приложения Redmine для автоматизации учета заявок на ремонт и обслуживание техники, результаты проекта его внедрения, а также ключевые положения регламента по сопровождению данного программного решения.

Ключевые слова: сопровождение, веб-приложение Redmine, внедрение, техническая инфраструктура.

Задача грамотной организации процессов внедрения и сопровождения программного продукта – это работа сотрудников ИТ-подразделения компании. Исследование деятельности отдела технической инфраструктуры компании ООО «Корпоративные системы Плюс» позволило определить, что ввод в действие веб-приложения

«Redmine» для автоматизации учета заявок на ремонт и обслуживание офисной техники компании будет способствовать получению следующих бизнес-выгод:

- интеграции (улучшению взаимодействия) отделов компании с отделом технической инфраструктуры (ТИ);
- консолидированию информации для принятия управленческих решений по оптимизации процессов управления технической и информационной инфраструктурой компании;
- уменьшению среднего рабочего время каждого сотрудника отдела ТИ на выполнение заявки;
- увеличению числа выполненных заявок отдела ТИ по обслуживанию технической и информационной инфраструктурой компании.

В результате реализации проекта внедрения были получены следующие его продукты:

- настроенный сервер для веб-приложения «Redmine», настройка восьмидесяти рабочих станций;
- пользователи, прошедшие инструктаж по работе в веб-приложении «Redmine»;
- своевременное оповещение сотрудников ТИ отдела о случившейся поломки (неисправности), проблемах;
- автоматизированный ввод, хранение и просмотр информации о поступивших заявках;
- оповещение об изменениях с помощью RSS-потоков и электронной почты;
- реализованная возможность просмотра текущих результатов о выполнении заявки;
- регламент сопровождения;
- акт ввода в ОПЭ;
- проектная документация (Техническое задание, план-график и т.д.).

В целях обеспечения сотрудника отдела технической инфраструктуры (администратора внедренного программного решения) четкими рекомендациями по осуществлению процесса сопровождения веб-приложения Redmine был составлен его регламент.

Регламент представляет не только описание процесса обслуживания и сопровождения веб-приложения Redmine (далее Системы), но и устанавливает перечень ролей и функциональные обязанности пользователей Системы, а также ответственность пользователей за работу в Системе.

Характеристика объекта обслуживания определяется представлением его функциональных возможностей, характеристикой программного и аппаратного обеспечения.

Функциональные возможности:

- ведение нескольких проектов;
- гибкая система доступа, основанная на ролях;
- система отслеживания ошибок;
- создание диаграммы Ганта и календаря;
- ведение новостей проекта, документов и управление файлами;
- оповещение об изменениях с помощью RSS-потоков и электронной почты;
- вики для каждого проекта;
- форумы для каждого проекта;
- учёт временных затрат;
- настраиваемые произвольные поля для инцидентов, временных затрат, проектов и пользователей;
- интеграция с системами управления версиями (SVN, CVS, Git, Mercurial, Bazaar и Darcs);
- создание записей об ошибках на основе полученных писем;
- поддержка множественной аутентификации LDAP;
- самостоятельная регистрация новых пользователей;
- многоязычный интерфейс (в том числе русский);
- поддержка СУБДMySQL, Microsoft SQL Server, PostgreSQL, SQLite, Oracle.

Комплекс технических средств Системы содержит: сервер, на котором установлена Система; локальная вычислительная сеть и доступ в Интернет; клиентские рабочие станции с доступом в Интернет; источники бесперебойного питания для обеспечения надежной работы сервера при сбоях и выключениях электропитания.

Программное обеспечение серверной части:

- операционная система: Windows Server 2012 R2.;
- дистрибутив продукта (installer bitnami Redmine 3.2.0);
- дополнительное программное обеспечение: антивирусное программное обеспечение.

Программное обеспечение клиентских станций:

- операционная система: Unix, Linux, Mac и Windows
- один из нижеперечисленных web-браузеров: MS Internet Explorer не ниже 10.0 версии; Firefox не ниже 28.0 версии; Google Chrome не ниже 34.0.1847.131 версии; Safari не ниже 7.0 версии; Opera не ниже 18.0.1284.68 версии;

– дополнительное программное обеспечение: антивирусное программное обеспечение; MS Office не ниже 2007.

Процесс сопровождения Системы должен предусматривать: передачу практических навыков применения Системы конечным пользователям; оказание технической и методологической помощи при освоении новой версии. С технологической точки зрения процесс сопровождения рассматривается как бизнес-процесс, ответственность за выполнение которого несет сотрудник отдела технической инфраструктуры (табл. 1).

Таблица 1

Основные функции сотрудника отдела технической инфраструктуры по осуществлению процесса сопровождения веб-приложения Redmine

Функция	Результат	Периодичность
Работы по обслуживанию, оптимизации производительности и настройке БД Системы	Увеличение производительности Системы	Раз в месяц
Организация и сопровождение резервного копирования БД Системы	Сохранение или восстановление информации в Системе	Каждый день
Управление доступом к ресурсам путем разграничения прав доступа и управления учетными записями	Создание или удаление проектов, задач, сотрудников, распределение ролей в проектах	Каждый день
Идентификация и аутентификация пользователей для доступа к информационным ресурсам средствами ОС	Безопасность прав пользователя Системой	Раз в две недели

Кратко представим технологию осуществления ряда ключевых функций. Так в рамках выполнения обязательств по обеспечению бесперебойного функционирования Системы исполнитель должен выполнить следующие работы:

– восстановление работоспособности системы в случае нештатных ситуаций;

- переустановка и настройка новых версий Системы;
- работы по обслуживанию, оптимизации производительности и настройке базы данных Системы;
- организация и сопровождение резервного копирования базы данных Системы.

Защита БД от несанкционированного доступа должна обеспечиваться: аутентификацией пользователя при входе в Систему; разграничением прав доступа к объектам Системы в соответствии с реализованной ролевой моделью.

Восстановление БД производится стандартными средствами сервера БД и предполагает следующий порядок работ:

- восстановить последнюю копию полного бэкапа БД;
- проверить целостность БД стандартными средствами сервера БД;
- убедиться в работоспособности Системы.

Файлы базы данных Системы должны находиться на устройствах хранения с высокой степенью отказоустойчивости, защищенных от перебоев питания в электрической сети управляемыми источниками бесперебойного питания. Ресурс источника питания должен обеспечить автономную работу, достаточную по времени для оповещения ответственного и корректного выключения сервера.

Все файлы базы данных подлежат обязательному включению в систему резервного копирования. Резервное копирование должно выполняться не реже, чем 1 раз в сутки в период наименьшей активности пользователей. Файлы с резервной копией должны находиться на физическом носителе, отличном от носителя оригинала базы данных. Желательно расположение его в другом помещении. Ответственный за резервное копирование обязан ежедневно просматривать отчеты о создании копий и незамедлительно реагировать на сообщения о сбоях, устранять неполадки.

Хранению подлежат следующие копии БД со сроком давности: 3 месяца; 1 месяц; 1 неделя – за каждый день; актуальная БД. Таким образом, для восстановления должны быть доступны 10 вариантов БД.

Организация работ по обновлению версий Системы: 1) исполнитель уведомляет сотрудников о доступности обновленной версии Системы посредством направления электронного письма; 2) исполнитель проводит консультации по функционированию Системы и по вопросам, связанным с переходом на модернизированную версию продукта, для специалистов и иных пользователей Системы.

Система поддерживает следующие режимы работы:

- нормальный режим работы – все устройства работают корректно в соответствии с правилами проверки функционирования;

– режим пуска/остановка – особый режим работы, инициируемый ответственными за обслуживание Системы. Характеризуется большой вероятностью нарушения корректной работы Системы и соответствующими подготовительными мерами;

– аварийное отключение – режим внезапного выхода из строя программной или аппаратной составляющей Системы.

Каждый режим работы Системы предполагает свой набор действий по сопровождению сотрудника отдела технической инфраструктуры.

В нормальном режиме функционирования проводятся следующие работы: ежедневный мониторинг системы; работы по обслуживанию БД; профилактические работы по поддержанию работоспособности рабочего места пользователя (по мере необходимости).

Пуск/остановка Системы предполагает выполнение плановых работ по внесению изменений в серверную часть и проводится в часы отсутствия активности пользователей с обязательным предварительным уведомлением о времени и продолжительности проведения работ.

Порядок проведения работ:

1) отключение общего доступа пользователей к серверной части (выполняется средствами операционной системы);

2) резервное копирование базы данных на отдельный носитель (при необходимости резервное копирование системных файлов Системы и другой критически важной информации (выполняется средствами операционной системы и/или ПО резервного копирования));

3) отключение серверной части при необходимости внесения изменений в аппаратную часть.

4) проведение плановых работ (изменение программной или аппаратной части);

5) проверка правильности функционирования серверной части;

6) запуск Системы и проверка основных режимов работы Системы;

7) подключение пользователей к работе с Системой (выполняется средствами операционной системы).

Аварийное отключение оборудования. Работы по восстановлению работоспособности Системы проводятся в оперативном режиме и могут включать в себя следующие мероприятия:

– восстановление сбоев аппаратной части (замена жестких дисков, оперативной памяти и т.д.);

– восстановление операционной системы;

– восстановление серверной части Системы;

– восстановление базы данных Системы.

Точный список мероприятий определяется в конкретной ситуации в зависимости от характера аварии.

Опыт применения таких регламентированных процедур не позволяет пока судить о высокой степени эффективности предлагаемой технологии. Те не менее их использование в течение двух лет сделало возможным быстрое и грамотное реагирование сотрудников технической инфраструктуры не только на ситуации, связанные с сопровождением самой Системы, но и более эффективное выполнение заявок по ремонту и обслуживанию аппаратно-программного парка компании в целом.

Список используемых источников

1. Масленникова О.Е. Актуальность обеспечения информационной безопасности при модернизации корпоративных информационных систем/О.Е. Масленникова // Новые информационные технологии в образовании. материалы VII междунар. науч.-практ. конф.. Рос. гос. проф.-пед. ун-т. Екатеринбург, 2014. С. 519-523.

2. Масленникова О.Е. Модернизация корпоративной информационной системы: от понятия к модели процесса//Перспективное развитие науки, техники и технологий: материалы 3-й Междунар. научн.-практ. конф. (18 октября 2013 года). - Курск: Изд-во: Юго-Зап. гос. ун-т. В 3 томах, 2013. -Том 2. -С. 307-312.

3. Масленникова О.Е., Назарова О.Б. Типовой проект внедрения корпоративной информационной системы для строительных организаций/

О.Е. Масленникова, О.Б. Назарова//Электротехнические системы и КОМПЛЕКСЫ. -2015. № 2 (27). С. 47-52.

4. Масленникова О.Е. Типовой проект внедрения корпоративной информационной системы на крупное промышленное предприятие/О.Е. Масленникова// Актуальные проблемы современной науки, техники и образования.- 2015. Т. 2. № 1. С. 149-152.

Обзор утилит для администрирования вычислительных систем на платформе Linux

*Кориунов Э.Н., Абдрахманов Д.В.,
ФГБОУ ВО «Магнитогорский государственный технический
университет им. Г.И. Носова»*

Аннотация. В статье представлен подробный обзор популярных утилит для администрирования вычислительных систем на платформе

Linux. Приводится двухуровневая классификация инструментальных средств.

Ключевые слова: утилиты, администрирование, классификация

Ежедневный труд системных администраторов, является неотъемлемой частью поддержки функционирования ИТ-инфраструктуры, обеспечению бесперебойного, своевременного доступа к сервисам и системам.

Для обеспечения стабильной работы ИТ-инфраструктуры любого масштаба, необходимо тщательно планировать политику резервного копирования, стратегии ликвидации различных сбоев(в том числе стихийных бедствий), плановое техническое обслуживание, аудит безопасности и т.д. [5].

В настоящее время существует широкий спектр специализированного программного обеспечения для администрирования вычислительных систем.

При должном подходе и использовании «правильного» инструментария, системный администратор способен сохранить работоспособность ИТ-инфраструктуры с минимальным временем простоя критически важных служб и систем, максимизируя тем самым время бесперебойной работы.

В данной статье мы рассмотрим утилиты для администрирования вычислительных систем на базе Linux.

Весь рассматриваемый инструментарий классифицируется на два обширных класса:

1. CLI - интерфейс командной строки (англ. Command Line Interface). Common Language Infrastructure

2. GUI – графический интерфейс пользователя (ГИП), графический пользовательский интерфейс (ГПИ) (англ. graphical user interface)

Внутри выделенных классов, все утилиты делятся по функциональным возможностям на 9 групп:

1. Сетевые утилиты
2. Средства безопасности
3. Инструменты управления накопителями
4. Инструменты обработки журналов
5. Инструменты резервного копирования
6. Инструменты мониторинга производительности
7. Инструменты производительности
8. Инструменты управления пакетами
9. Аппаратные средства

В среде профессиональных администраторов наибольшей популярностью пользуются утилиты с CLI интерфейсом. Это обусловлено рядом преимуществ:

- Низкие требования к аппаратным средствам
- Высокая степень унификации
- Широкая возможность интеграции программ

Недостатки командного интерфейса:

- Плохая наглядность интерфейса
- Ограниченные возможности вывода информации.

Рассмотрим наиболее популярные CLI утилиты из представленных категорий.

Сетевые утилиты

1. **Ping**: проверка соединения с удаленным хостом посредством протокола ICMP.

2. **Hping**: позволяет генерировать специальные ICMP/UDP/TCP пакеты и просматривать ответы пингуемого хоста в стиле обычной утилиты ping. HPING поддерживает фрагментацию, позволяет определять размер и содержимое тела пакета и может использоваться для передачи файлов по поддерживаемым протоколам.

Средства безопасности

1. **Iptables** - утилита является стандартным интерфейсом управления работой межсетевого экрана для настройки брандмауэра ядра Linux. Предоставляет средства для создания и изменения правил для входящих, исходящих и транзитных пакетов.

2. **Nmap** - популярный инструмент сканирования портов и обнаружения сети, используются для целей аудита безопасности [3, 4].

Инструменты управления накопителями

1. **FDISK** - общее название класса утилит для управления разделами жесткого диска. Используется для просмотра, создания и редактирования разделов диска на жестких дисках и съемных носителях.

2. **Sfdisk** - вариант FDISK, который осуществляет доступ или обновляет таблицу разделов в неинтерактивном режиме. Используется для автоматизации разбиения диска на разделы как часть резервного копирования и восстановления.

Инструменты обработки журналов

1. **Tail** - используется для вывода части лог-файла (как правило последние 10 строк). Другой вариант multital (мониторинг нескольких файлов) и ztail (поддержка регулярных выражений, фильтрации).

2. **Logrotate** - инструмент позволяет разделять, сжимать и отправлять на электронную почту старые / большие файлы журналов в

заранее определенном интервале. Полезно для администрирования высоконагруженных серверов, которые могут производить большой объем лог-файлов.

Инструменты резервного копирования

1. **Rsync** - выполняет синхронизацию файлов и каталогов в двух местах с минимизированием трафика, используя кодирование данных при необходимости. Часто используется для репликации хранилища данных на внешнее хранилище, при необходимости через защищенное соединение, такие как SSH или Stunnel.

2. **RDIFF-backup** - эффективный инструмент резервного копирования. Основной идеей rdiff-backup является сочетание преимуществ полного и инкрементального копирования. Приложением сохраняются различия текущей и предыдущей копии исходного каталога в отдельности, поэтому имеется возможность восстановления давно утерянных файлов [1].

Инструменты мониторинга производительности

1. **Top** - утилита мониторинга запущенных процессов. Может контролировать загрузку системы, состояния процесса, использования ресурсов процессора и памяти (Htop разновидность утилиты top с расширенными возможностями).

2. **PS** - показывает снимок всех запущенных процессов в системе. Выход может отражать PID, PPID, пользователя, нагрузку, память, общее время пользователя / системы, время начала, и многое другое. Варианты включают pstree, который показывает процессы в древовидной иерархии.

Инструменты производительности

1. **Screen** - утилита-мультиплексор, предоставляющая пользователю доступ к нескольким сессиям в рамках одной сессии.

2. **Cheat** - утилита ищет установленные в файлах базы данных краткие описания системных команд по ключевому слову и выводит результат на стандартный вывод

Инструменты управления пакетами

1. **APT** - де-факто менеджер пакетов для систем на базе Debian, таких как Debian, Ubuntu.

2. **APT-fast** - утилита ускоряет загрузку новых пакетов, делая это в несколько потоков.

Аппаратные средства

1. **Lspci** - утилита позволяет выводить различную информацию об установленных устройствах PCI, таких как имена моделей, драйверы устройств, возможности, адреса памяти, шины PCI-адреса.

2. **Lshw** - утилита отображает подробную информацию о конфигурации аппаратных средств различных категорий (например, процессор, память, материнская плата, сеть, видео, хранение).

GUI утилиты

MySQL Workbench является одним из универсальных инструментов для работы с базами данных MySQL. Работа осуществляется как локально, так и удаленно, с хорошо разработанным графическим интерфейсом.

MySQL Workbench предназначен не только для управления ранее созданными базами данных, но так же осуществлять возможность разрабатывать, проектировать и администрировать новые базы данных MySQL.

Расширение функциональных возможностей MySQL Workbench за счет использования дополнений позволяет интегрировать данный инструмент с таким программным обеспечением как Microsoft SQL Server, Microsoft Access, Sybase ASE, PostgreSQL. Первая версия MySQL Workbench была выпущена в сентябре 2005 года. Разработчиком является Oracle Corporation.

PhpMyAdmin является еще одним инструментом администрирования MySQL. С помощью PhpMyAdmin администратор может создавать базы данных MySQL, а также управлять ими через стандартный веб-браузер.

Приложение пользуется большой популярностью у веб-разработчиков, так как позволяет управлять СУБД MySQL без непосредственного ввода SQL команд, предоставляя дружелюбный интерфейс. Подавляющее большинство российских провайдеров используют это приложение в качестве панели управления для того, чтобы предоставить своим клиентам возможность администрирования выделенных им баз данных.

Webmin представляет собой программный комплекс, позволяющий администрировать операционную систему через веб-интерфейс, в большинстве случаев позволяя обойтись без использования командной строки и запоминания системных команд и их параметров.

Используя любой браузер, администратор сервера может создавать новые учётные записи пользователей, почтовые ящики, изменять настройки служб и сервисов, например, веб-сервера Apache, DNS. Однако, в некоторых случаях необходимо знание операционной системы и редактирование конфигурационных файлов вручную. Кроме того, не все возможности операционной системы и не все программы можно конфигурировать через интерфейс Webmin, например, nginx пока не входит в базовый набор. Разработчиком является Jamie Cameron и сообщество Webmin.

YaST программа, используемая в openSUSE и SUSE Enterprise Linux для установки системы и ее администрирования после установки. Пользуется популярностью за простоту в использовании, привлекательный графический интерфейс и возможность настроить

систему быстро после установки. YaST - это еще один Инструмент Настройки (Yet another Setup Tool).

YaST может использоваться для настройки всей системы. Разработчик – Novell.

Shorewall представляет собой графический интерфейс для конфигурирования IPTables (утилита командной строки, является стандартным интерфейсом управления работой межсетевого экрана брандмауэра).

С помощью данного графического интерфейса безопасности, можно настроить шлюзы, виртуальные частные сети, управление трафиком, черный список, и многое другое. Первая версия Shorewall появилась в 1999 году. Основной разработчик — Томас Истеп (Thomas M.Eastep).

Apache Directory является открытым исходным проектом Apache Software Foundation. Ее флагманский продукт, Apache Directory Server, это встраиваемый сервер каталогов полностью написан на Java.

Продукт был сертифицирован LDAPv3-совместимой с The Open Group в 2006 году. Кроме LDAP, сервер поддерживает другие протоколы, а также, и сервер Kerberos.

Подпроект реализует инструмент каталога Eclipse, на основе Apache под названием DirectoryStudio. Она включает в себя LDAP-браузер / редактор, браузер схемы, редактор LDIF, редактор DSML, и многое другое.

CUPS (CommonUNIXPrintingSystem) — сервер печати для UNIX-подобных операционных систем. Компьютер с запущенным сервером CUPS представляет собой сетевой узел, который принимает задания на печать от клиентов, обрабатывает их и отправляет на соответствующий принтер.

CUPS имеет встроенный веб-интерфейс администрирования, который работает через IPP. Для вызова панели администрирования обычно используется URL <http://localhost:631>.

Таким образом, администрирование может осуществляться средствами любого веб-браузера. Однако, существует также множество графических средств настройки CUPS. Например, среда GNOME содержит утилиту `gnome-cups-manager`, позволяющую администрировать CUPS, а в среде KDE используются средства администрирования, специфичные для различных дистрибутивов.

Разработка CUPS началась в 1997 году, а первая публичная бета-версия появилась через два года. Разработчиком является Майкл Р. Свит.

Zenmap - официальный GUI для программы NmapSecurityScanner. Программа написана на языке Python и работает на платформах, на которых работают Python и nmap.

Распространяется на условиях свободной GNU General Public License. Часто используемые настройки сканирования могут быть сохранены как профили для повторного использования.

Мастер создания команд поддерживает интерактивное создание командных строк Nmap. Сканированные результаты сохраняются и могут быть просматриваться позже, сравниваться с другими результатами на факт изменений. Результаты сохраняются в поисковой базе данных.

Cockpit был создан RedHat, чтобы сделать администрирование сервера проще. С помощью этой веб-интерфейс администратор может решать такие задачи, как администрирование системы хранения, инспекции журнала, запуск / остановка служб, а также мониторинг нескольких серверов.

Cockpit работает на таких системах как Fedora Server, Arch Linux, CentOS, Fedora Atomic Atomic и Red Hat Enterprise Linux.

Таким образом, подводя итог по GUI инструментам, можно отметить большое разнообразие программных средств, обладающих не только гибким графическим интерфейсом, но и широкими функциональными возможностями.

В завершении отметим, что в статье рассмотрена незначительная часть доступного в настоящее время программного обеспечения для администрирования вычислительных систем на базе Linux.

Список использованных источников

1. RDIFF-backup [Электронный ресурс]. - Режим доступа: <http://www.securitylab.ru/software/233360.php>

2. Zenmap [Электронный ресурс]. - Режим доступа: <http://zenway.ru/page/rdiff-backup>

3. Гаврилова И.В. Организация защиты персональных данных в образовательных учреждениях // Новые информационные технологии в образовании Материалы VII международной научно-практической конференции. Российский государственный профессионально-педагогический университет. Екатеринбург, 2014. С. 509-513.

4. Гаврилова И.В., Гаврилов Д.О. Настройка контент-фильтрации в образовательных учреждениях // Новые информационные технологии в образовании Материалы VI Международной научно-практической конференции. 2013. С. 317-319.

5. Рубан К.А. Разработка методики построения открытой образовательной среды / К.А. Рубан// Вестник компьютерных и информационных технологий. 2010. № 6. С. 53-56.

**Использование свободного программного обеспечения в отделе АСУ
«Транспорт» ООО «ММК-Информсервис»**

Аннотация. В статье приводится описание программной составляющей информационной инфраструктуры ООО «ММК-Информсервис». Сделан акцент на простоте установки и эксплуатации свободно-распространяемых пакетов без ущерба производительности локальной вычислительной сети организации.

Ключевые слова: свободное программное обеспечение, «ММК-Информсервис», администрирование, программирование, проектирование.

ООО «ММК-Информсервис» образовано в мае 2010 года на базе управления информационных технологий ОАО «ММК» [1]. Общество занимается обеспечением текущих и перспективных потребностей ОАО «ММК» и Группы обществ ОАО "ММК" в полноценной, оперативной и достоверной информации с целью повышения качества принимаемых управленческих решений.

Одним из подразделений ООО «ММК-Информсервис» является группа техподдержки и развития АСУ «Транспорт». Отдел АСУ «Транспорт» образовано в первой половине 80-х годов, занимается сопровождение и разработкой программного обеспечения для функционирования и развития железнодорожного транспорта ОАО «ММК». В отделе работает порядка 20 человек: электромеханики, инженеры-электроники, инженеры-программисты. Сопровождение и работа с пользователями ведется круглосуточно, часть персонала работает железнодорожному графику, что обеспечивает оперативное решение возникающих проблем.

Производится работа с пользователями всех категорий и должностей, начиная от приёмосдатчика груза и багажа, заканчивая начальниками цеха. Решаются задачи разного уровня сложности, от простых, таких как установка программного обеспечения, настройка профиля пользователя, подготовка компьютера или консультация по работе с АРМами, до сложных и более трудоёмких, требующих разработки нового документа, сбора и анализа информации за большой промежуток времени, внедрения нового программного продукта. Все эти задачи решаются максимально оперативно и качественно, персонал в отделе опытный и высокообразованный.

Для эффективной работы, как самого отдела, так и обслуживаемых им подразделений, необходимо разнообразное и сложное программное обеспечение (ПО). Лицензионные версии ПО зачастую достаточно дорогостоящие, и их приобретение в необходимом количестве ограничено выделяемым бюджетом. В этом случае в качестве

альтернативы рассматривается свободное программное обеспечение (СПО) [2]. В настоящее время имеются свободные аналоги практически любого коммерческого продукта и зачастую эти продукты лучше, чем коммерческие продукты. Поэтому в отделе АСУ «Транспорт» широко используются такие программы.

Отдел эксплуатации использует:

- **7zip** – свободный файловый архиватор с высокой степенью сжатия данных, является аналогом WinZip. Множество поддерживаемых форматов, возможность создания самораспаковывающихся архивов, поддержка 64-разрядной операционной системы Windows.

Назначение: используется повсеместно для стандартной подготовки компьютера к работе.

- **Foxit Reader** – бесплатное прикладное программное обеспечение для просмотра электронных документов в стандарте PDF для операционных систем Windows, является альтернативой пакету AdobeReader, по сравнению с которым имеет меньший объем, меньшую требовательность к ресурсам и более высокое быстродействие.

Назначение: так же как и 7zip используется для стандартной подготовки компьютера к работе.

- **Far** – консольный файловый менеджер для операционных систем семейства Microsoft Windows. К утилите можно подключать огромное количество плагинов. Позволяет с большим удобством и комфортом оперировать файлами, находящимися на жестком диске компьютера и всевозможных носителях информации (флешках, дисках, жестком диске). Кроме того, есть возможность работать с FTP-серверами и управлять принтерами.

Назначение: используется для администрирования рабочих мест пользователей, в основном используется Portable версия.

- **Apache OpenOffice** (ранее OpenOffice.org, OO.org, OO.o, OOo) — свободный пакет офисных приложений. Конкурирует с коммерческими офисными пакетами как на уровне форматов, так и на уровне интерфейса пользователя. Одним из первых стал поддерживать новый открытый формат OpenDocument. Официально поддерживается на платформах Linux, Microsoft Windows, Mac OS X Intel/PowerPC.

Назначение: устанавливается на компьютеры дочерних организаций, где нет основной лицензии для MSOffice. Вполне удовлетворяет потребности обычного пользователя, удобен и прост в освоении, обладает небольшим размером пакета.

- **hMailServer** — бесплатный почтовый сервер под платформу Windows, созданный Martin Knaflve. Он работает как служба Windows и включает в себя инструменты администрирования и резервного копирования. Поддерживает почтовые протоколы IMAP, POP3 и SMTP.

Для хранения настроек и индексов использует базы данных типа MySQL, MS SQL или PostgreSQL, сами же почтовые сообщения хранятся на жестком диске в формате MIME.

Назначение: в АСУ «Транспорт» используется на серверах для передачи логов, обмена информацией. Была основным почтовым сервером ЖДТ до перехода предприятия на Microsoft Exchange Server

Отдел разработки использует:

- **GIMP** (GNU Image Manipulation Program, «Гимп») — свободно распространяемый растровый графический редактор, программа для создания и обработки растровой графики и частичной поддержкой работы с векторной графикой. Распространяется на условиях GNU General Public License.

Назначение: на рабочих местах используется как более мощная и функциональная альтернатива Paint. Не обязателен для установки при подготовке ПК, устанавливается по просьбе пользователей. Также используется инженерами-программистами для оформления web страниц дизайна программных продуктов.

- **Notepad++** — свободный текстовый редактор с открытым исходным кодом для Windows с подсветкой синтаксиса большого количества языков программирования и разметки. Поддерживает открытие более 100 форматов. Базируется на компоненте Scintilla, написан на C++ с использованием STL, а также Windows API и распространяется под лицензией GNU General Public License. Базовая функциональность программы может быть расширена как за счёт плагинов, так и сторонних модулей, таких как компиляторы и препроцессоры.

Основными возможностями Notepad++ являются: авто завершение набираемого слова, возможность создания собственного списка API функций, перетягивание фрагментов текста, автоматическое определение состояния файла, запись макроса и его выполнение, также многое другое, все это хорошее подспорье для разработчика и инженеров отдела.

Назначение: используется инженерами-программистами для облегчения работы с основными программными продуктами.

- **jEdit** это кроссплатформенный редактор для программистов, написанный на языке Java. Он поддерживает большое количество языков программирования, такие как C, C++, C#, python, pascal и многие другие и разметки. Редактор легко расширяется с помощью встроенного в него скриптового языка BeanShell, на котором можно писать макрокоманды (макросы), позволяющие существенно экономить время и облегчать процесс редактирования. Помимо этого, для jEdit написано достаточно большое количество расширений, которые могут быть легко установлены

прямо из редактора. Плагины значительно расширяют возможности jEdit, позволяя, например, интегрировать внешние утилиты, такие как CTags.

Назначение: как и Notepad++ используется инженерами-программистами для облегчения работы с основными программными продуктами.

- **FreeMind** — свободная бесплатная программа для создания диаграмм связей. FreeMind написана на Java и распространяется согласно GNU General Public License. К возможностям программы относятся: наглядное представление информации, поддержка импорта и экспорта информации, наглядное получение карт, скрытие ветвей, совместное редактирование по сети. Программа обладает расширенными возможностями экспортирования, легка в освоении для любой категории пользователей.

Назначение: Используется инженерами-программистами при разработке ПО.

- **Inkscape** (Инкскейп) — свободно распространяемый векторный графический редактор, удобен для создания как художественных, так и технических иллюстраций. Это стало возможным во многом благодаря открытому формату SVG, развиваемому консорциумом W3C. Формат SVG позволяет создавать иллюстрации различного типа, в том числе анимированные. К удобству использования относятся: привычные инструменты работы с графическими редакторами, контекстная панель параметров инструментов, большое количество продуманных клавиатурных комбинаций, информативная статусная строка, встроенный векторизатор растровых изображений, коллективное рисование через протокол XMPP.

Назначение: используется разработчиками графической обработки и наведения окончательных штрихов рисунка.

- **ProjectLibre** - кроссплатформенное программное обеспечение для управления проектами. Распространяется на условиях лицензии Common Public Attribution License Version 1.0 (CPAL). Позиционируется создателями как открытая замена коммерческому продукту Microsoft Project. Доступно для операционных систем Microsoft Windows, Linux, Unix, Mac OS X. К возможностям программы относятся: поддержка форматов файлов MicrosoftProject 2010, интерфейс пользователя Ribbon, поддержка диаграмм Ганта, удобство использования и освоения продукта.

Назначение: используется инженерами-программистами для разработки и составление проектов по внедрению или разработки программных продуктов.

- **NextGIS QGIS** — полнофункциональная настольная геоинформационная система (ГИС), основанная на QGIS, предназначена

для создания и редактирования данных, производства карт, выполнения аналитических операций. QGIS активно развивается международным сообществом разработчиков, в котором Россия представлена командой NextGIS. К возможностям относятся: просмотр векторных и растровых данных в различных форматах и проекциях, удобный графический интерфейс для создания карт, возможность создавать и редактировать, векторные данные, экспорт данных в различные форматы, анализ векторных и пространственных данных, публикация карт в сети Интернет.

Назначение: В АСУ «Транспорт» используется разработчиками для отображения на ресурсах сайте железнодорожного транспорта ОАО «ММК» движение поездов, занятость тупиков вагонами.

• **SharpDevelop** — свободная среда разработки для C#, Visual Basic.NET, Boo, IronPython, IronRuby, F#, C++. Обычно используется как альтернатива Visual Studio .NET. Существует также форк на Mono/GTK+ — MonoDevelop. SharpDevelop предоставляет интегрированный отладчик, который использует собственные библиотеки и взаимодействует с исполняющей средой .NET через COM Interop. Хотя SharpDevelop использует файлы проекта в формате MSBuild, он по-прежнему может использовать компиляторы от .NET Framework 1.0 и 1.1, а также от Mono.

Назначение: основная среда для разработки инженеров-программистов.

• **Microsoft Visual Studio Express** — линейка бесплатных интегрированных сред разработки, облегчённая версия Microsoft Visual Studio, разработанной компанией Microsoft. Согласно утверждению Microsoft [3], «Express»-редакции предлагают отлаженную, простую в обучении и использовании среду разработки пользователям, не являющимся профессиональными разработчиками ПО, — любителям и студентам. С апреля 2009 года Microsoft прекратила поддержку всех предыдущих версий Visual Studio Express.

Назначение: одна из используемых сред разработки в отделе.

• **FlashDevelop** — свободная среда разработки и редактор, написанный на C# и использующий для вывода текста компоненту Scintilla, позволяющую создавать Flash-приложения при помощи Flex SDK, MTASC или haxe. Является одной из альтернатив Flash Builder для разработки веб-приложений (благодаря своей бесплатности и заметно большему быстродействию) хотя и не обладает всеми возможностями последнего. Поддерживает профилирование, отладку Adobe Flex и имеет умное автодополнение при написании кода на языке ActionScript. Также при использовании Adobe Flex SDK 4.5.1 можно писать приложения для платформ Android, iOS.

Назначение: одна из используемых сред разработки в отделе.

При подготовке рабочего места в среднем используется восемь программных продуктов (Антивирус Касперский, Far, Unicenterremotecontrol, Гранд-Смета, FoxitReader, АРМ (разные), 7zip, MSoffice), из них три являются СПО, что составляет 37.5 %.

Процесс разработки прикладного программного обеспечения можно полностью покрыть использованием свободного ПО. Коммерческими остаются только программы для работы с базами через MicrosoftSQL. Недавно закончился переход на свободную систему управления версиями и проектами Git (<https://about.gitlab.com>).

Свободное программное обеспечение играет большую роль в работе отдела, без его использования сложно представить полноценное и эффективное выполнение производственных задач.

Список использованных источников

1. Официальный сайт ООО «ММК-информсервис» [Электронный ресурс] – Режим доступа: [http://www/ http://is-mmk.ru/](http://www/http://is-mmk.ru/)
2. Свободная энциклопедия [Электронный ресурс] – Режим доступа: <https://ru.wikipedia.org/>
3. Официальный сайт Майкрософт [Электронный ресурс] – Режим доступа: <http://www.microsoft.com/ru-ru/>

Опыт внедрения свободного программного обеспечения виртуализации серверной инфраструктуры организации(на примере МГТУ им. Г.И. Носова)

*Рубан К. А., к.т.н.,
ФГБОУ ВО «Магнитогорский государственный технический
университет им. Г.И. Носова»*

Аннотация. В статье представлен опыт решения проблемы виртуализации серверной инфраструктуры с помощью свободного программного обеспечения.

Ключевые слова: Python, Network UPS Tools

В связи с необходимостью внедрения различных сервисов и поддержки большого спектра программного обеспечения, используемого в учебном процессе, доля серверов в образовательных учреждениях с каждым годом возрастает, а стоимость их программного и аппаратного обеспечения нередко превышает десятки миллионов рублей. Сегодня виртуализация серверной инфраструктуры предприятия является не только средством экономии и рационального использования

вычислительных ресурсов, но и возможностью оптимизировать затраты на администрирование и сопровождение серверов.

Рынок решений в области виртуализации серверной инфраструктуры достаточно насыщен и представлен как проприетарным, так и свободным программным обеспечением (СПО). Как правило, выбор конкретного средства зависит от нескольких факторов. В процессе анализа для МГТУ были выделены следующие ключевые требования:

- минимизация совокупной стоимости владения;
- поддержка аппаратной виртуализации – HVM (в т.ч. на базе Intel VT);
- поддержка паравиртуализации – PV;
- возможность установки широкого спектра гостевых операционных систем (Linux, Unix, Windows, OS X);
- централизация управления ресурсами;
- стабильность, безопасность и сопровождаемость.

Основным преимуществом СПО принято считать его бесплатность, что при прочих равных позволяет снизить совокупную стоимость владения инфраструктурой. При этом зачастую не рассматривается такой важный аспект как открытость, а, соответственно, и модернизируемость системы. Поскольку для получения дополнительных (иногда жизненно необходимых) функций для большинства проприетарных систем виртуализации требуется приобретение лицензий, открытость дает дополнительные преимущества, т.к. позволяет реализовать эти функции самостоятельно, без значительного увеличения стоимости всей системы.

На момент разработки проекта в распоряжении университета имелся следующий набор аппаратного обеспечения:

- шасси IBM PureFlex с 4мя blade-серверами (8 процессоров Intel Xeon 8C E5-2670 (128 ядер), 256Gb PC3-12800 ECC DDR3 1600MHz), 2мя коммутаторами сети хранения данных (Storage Area Network, SAN), 2мя коммутаторами 3го уровня для связи с внешней средой;
- система хранения данных (СХД) IBM v7000;
- SAN, построенная на базе указанных выше СХД и оптических коммутаторов, со скоростью обмена информацией между СХД и шасси 48Gb/s;
- СХД IBM v3700; коммутаторы, обеспечивающие связь шасси с корпоративной информационно-вычислительной сетью университета на скорости 20Gb/s.
- полностью зарезервированные компоненты шасси (модули управления, блоки питания, модули охлаждения и т.д.);

- два источника бесперебойного питания, подключенных к питающей линии с автоматическим выбором резерва (резервированное питание с двух подстанций).

Анализ указанных требований, имеющегося оборудования и систем виртуализации позволил остановить выбор на гипервизоре XEN, который обладает всеми необходимыми для проекта параметрами:

- свободно распространяемый исходный код;
- работа с SAN;
- поддерживает HVM и PV;
- поддерживает технологию Intel VT;
- поддерживает весь необходимый спектр гостевых операционных систем (в т.ч. Linux в режиме паравиртуализации).

Заметим, что при использовании проприетарного решения Citrix XenServer, которое базируется на свободном гипервизоре XenProject, организация получает дополнительные функции, часть из которых доступна бесплатно, а часть – за отдельную плату. В рамках рассматриваемого проекта функциональные возможности выбранного средства полностью удовлетворяли поставленным задачам, а дополнительные предполагалось реализовывать собственными силами.

Особой проработки требовала система мониторинга и автоматического отключения серверной фабрики, т.к. аварийное прекращение питания может привести не только к неконсистентности данных на виртуальных серверах, но и к проблемам с аппаратным обеспечением всей системы.

Для мониторинга состояния ИБП на базе СПО Network UPS Tools (NUT) был разработан скрипт на языке Python, который осуществляет опрос источников резервного питания и при достижении критического уровня заряда батарей (менее 30 минут автономной работы) инициирует процедуру выключения питания виртуальных машин (VM), СХД, хост-серверов, дополнительного оборудования и шасси. В случае, если на гостевой системе установлен набор расширений XenTools, то процедура выключения заменяется процедурой приостановки (suspend), что позволяет сохранить текущее состояние VM и существенно ускоряет её восстановление. Если виртуальная машина не может быть приостановлена, или процедура не была завершена в отведенный период времени, то к ней применяется процедура принудительного выключения. Если в процессе работы скрипта питание было восстановлено, то скрипт останавливает процедуру выключения фабрики и переходит в режим мониторинга.

Стоит отметить, что автору не удалось найти полноценных решений (проприетарных и свободных), которые реализовывали бы описанную выше задачу. В данном случае открытость API и применение СПО позволило самостоятельно реализовать необходимый функционал.

Таким образом, в течение года на серверную фабрику было импортировано 14 физических серверов и развернуто 17 новых VM, что позволило заменить 31 физический сервер 4мя blade-серверами. Кроме прямой финансовой выгоды, выбранный подход позволил существенно сэкономить на инфраструктуре центра обработки данных (система кондиционирования и питания), а также увеличить скорость установки и настройки новых серверов (шаблонизация).

Отметим, что текущая нагрузка на серверную фабрику составляет около 60% от заложенной проектной мощности, что оставляет достаточный запас вычислительных ресурсов и позволяет обеспечить отказоустойчивость в случае выхода из строя любого компонента системы.

Перспективы развития виртуализированной серверной инфраструктуры видятся в автоматизации процесса создания инкрементных снимков состояния виртуальных машин (snapshots) и резервирования образов дисков, использовании технологии heartbeat для достижения высокой степени доступности виртуализированных систем (High-Availability systems) и предоставлении облачных ресурсов по модели «платформа как услуга» (Platform as a Service, PaaS) сотрудникам и студентам университета.

Организация учебного процесса Уральского радиотехнического колледжа им. А. С. Попова с использованием свободного программного обеспечения

*Уймин А. Г., Мешков А. Д.,
ГАПОУ СО "Уральский радиотехнический колледж им. А.С. Попова"*

Аннотация. В статье приводится описание становления и развития информационной инфраструктуры Уральского радиотехнического колледжа на базе свободного программного обеспечения. Подчеркивается ведущая роль колледжа среди вузов в области использования свободного программного обеспечения. Обоснованы достоинства применения СПО.

Ключевые слова: свободное программное обеспечение, ИТ-инфраструктура, среднее профессиональное образование

Политическая и экономическая ситуация в нашей стране требует оптимизации затрат на ИТ инфраструктуру, а также отказа от программного обеспечения зарубежных производителей. При этом идет реформа по преобразованию учебных заведений, их укрупнению и консолидации ресурсов; среднее профессиональное образование и высшее образование уже не дополняют друг друга, как это было ранее, а являются прямыми конкурентами на рынке образовательных услуг. Уровень финансирования среднее профессиональное образование (СрПО) и высшее профессиональное образование (ВПО) отличается на порядки, но конечный заказчик, работодатель, уже видит, что студенты, качественно подготовленные в среднем звене, имеют больше практических навыков по работе с современным аппаратным и программным обеспечением, при более низкой их стоимости как специалистов. К сожалению, большинство образовательных учреждений поставлено на грань выживания и образовательный процесс проходит либо без должного технического оснащения, либо на эмуляторах и виртуализаторах. Современный студент может получить хорошую теоретическую базу дистанционно, в том числе и бесплатно, но невозможно в домашних условиях за приемлемые финансовые вложения развернуть качественные стенды.

«СПО для нищобродов!» - с этой фразы одного из представителей сообщества СПО в Екатеринбурге хотелось бы начать данную статью. Уральский радиотехнический колледж им. А.С. Попова (УРТК) находится в Екатеринбурге. УРТК потовит специалистов по 12 специальностям, из них 6 - по ИТ-направлению. До 2008 года в УРТК учебном заведении из свободного программного обеспечения (СПО) были шлюз и сайт на FreeBSD. В учебном процессе СПО не использовалось никак. В 2008 году начался перевод учебных материалов на Linux.

В колледже 19 компьютерных лабораторий. Парк персональных компьютеров (ПК) насчитывает около 500 шт. Парк серверов 20 шт. Обслуживанием лабораторий занимаются заведующие лабораториями, которые являются и преподавателями специальных дисциплин.

Первой лабораторией, переведенной на Linux, стала лаборатория программно-аппаратной защиты объектов сетевой инфраструктуры. Выбор дистрибутива, который окажется основным в лаборатории, был долгим и сложным. Был пройден путь от FreeBSD>PC-BSD>Debian>Fedora>AltLinux. У AltLinux подкупила простота

начального вхождения, качество документирования и форум, на котором действительно помогают. Начали работать с версией с AltLinuxШкольный Новый Легкий, сейчас УРТКработает на AltLinuxCentaurusP7 и ожидает скорейшего выхода 8 платформы.

В таблице 1 приведён перечень лабораторий и указан год перехода на СПО. Во всех лабораториях расположено по 12 рабочих мест студентов, кроме 304, где 27 рабочих мест.

Таблица 1

Перечень лабораторий

№	Название лаборатории	Год перехода	Преподают дисциплины, связанные с...	На рабочем месте студента
101	Сборки, монтажа и эксплуатации средств вычислительной техники.	2012	аппаратным обеспечением ПЭВМ. Установка, настройка системного ПО, развертывание служб и сервисов. Техническое обслуживание и ремонт СВТ.	ПЭВМ, принтер, сканер.
104	Программно-аппаратной защиты объектов сетевой инфраструктуры	2009	сетевыми операционными системами, информационной безопасностью.	2 ПЭВМ, на которых студенты развертывают несколько сетей в системах виртуализации
114	Сетевая академия Cisco.	2013	сетевыми технологиями в рамках CiscoCCNA.	ПЭВМ
305	Программного обеспечения компьютерных сетей. Авторизованный центр Cisco Academy	2015	основами компьютерной грамотности.	ПЭВМ
117	Лаборатория беспроводных технологий	2014	сетевыми технологиями: беспроводные	ПЭВМ, с дискретными беспроводными и

№	Название лаборатории	Год перехода	Преподают дисциплины, связанные с...	На рабочем месте студента
			сети.	проводными адаптерами и видеокартой Radeon для анализа защищенности беспроводных соединений. В стенд входят два беспроводных маршрутизатора и IP-камера.
304	Информатики. Программирования и баз данных.	2015	информатикой и программированием.	ПЭВМ

Есть участки, на которых нет возможности перейти на СПО и приходится использовать Windows. В связи с тем, что колледж обучает по заказу конкретных предприятий и организаций, состав прикладного программного обеспечения диктуют они. Используются такие пакеты как AltiumDesigner, AutodeskAutocad, MSOffice, 1С и т.д.

Стандартным набором дополнительного ПО в лабораториях является: Remmina, Firefox, Geany, iTALC, OpenSSH, PuTTY, VirtualBox. К нестандартному относятся такие пакеты как CiscoPacketTracer, LinSSID, Aircrack-ng, Wireshark, MetasploitFramework.

Сервисы на базе СПО, используемые для обслуживания образовательного процесса, приведены в таблице 2.

Таблица 2

Используемые сервисы

Сервис	ИТ-среда	Назначение
Система видеонаблюдения	Debian 8 + MotionEye	Обеспечение безопасности функционирования лабораторий, отслеживание инцидентов
Локальное зеркало репозитория Debian	Debian 7 + apt-mirror	Проведение лабораторных работ

Сервис	ИТ-среда	Назначение
Локальное зеркало репозитория AltLinux	Alt Linux P7 + sisyphus-mirror	Обслуживание ПЭВМ
Сервер сетевой загрузки	Debian 7 + tftpd-hpa + isc-dhcp-server + syslinux + nfs-kernel-server + smb + apache2	Обслуживание ПЭВМ
Вики-энциклопедия	Alt Linux P7 + Mediawiki	Создание и хранение документации
Сервер SNMP-мониторинга	Debian 8 + Zabbix	Мониторинг сетевого оборудования
Система дистанционного обучения	FreeBSD 7 + Moodle	СДО

Сервер сетевой загрузки позволяет производить загрузку следующих образов ОС – Debian 8, AltLinuxP7, Windows 7/8.1/2012R2, а так же системных и служебных утилит HDT, memtest86+, MSDaRT, MHDD.

Методическое обеспечение разрабатывается полностью самостоятельно т.к., с одной стороны, отсутствуют актуальные и практико-ориентированные разработки, а с другой - все лабораторные работы являются практическими кейсами, разрабатываемыми совместно с партнерами. Это трудоемкий и долгий процесс. Из-за того, что методические указания каждый семестр обновляются и дополняются, пока работа по получению грифа УМО не ведется.

Разработка рабочей тетради для выполнения лабораторных работ по МДК.03.01. «Техническое обслуживание и ремонт компьютерных систем и комплексов» для специальности среднего профессионального образования 230113 «Компьютерные системы и комплексы» получила признание на Областном конкурсе методической продукции среди педагогов профессиональных образовательных организаций Свердловской области, ноябрь 2014 г.; Всероссийский профессиональный конкурс методических разработок «Методический потенциал в современном профессиональном образовании», декабрь 2014 г. и т.д.

В настоящий момент ведется апробация методического пособия по МДК 02.01 «Программное обеспечение компьютерных сетей» для специальности среднего профессионального образования 09.02.02

«Компьютерные сети». В комплекс входят следующие лабораторные работы:

- 1) установка и базовая настройка сетевой операционной системы;
- 2) организация доступа сетевых ОС в сеть Интернет через шлюз (в этой же лабораторной работе рассматривается установка дополнительного ПО и развёртывание сервисов);
- 3) развёртывание файлового сервера (рассматриваются Samba и NFS);
- 4) развёртывание DHCP;
- 5) развёртывание DNS;
- 6) развёртывание LDAP;
- 7) развёртывание почтового сервера;
- 8) настройка iptables и прокси-сервера;
- 9) статическая и динамическая маршрутизация.

На рисунке 1 приведен пример схемы сети одной из лабораторных работ.

Это схема итоговой лабораторной работы, все сервисы настраиваются последовательно, т.е. сначала разворачивается минимальная сеть, в которую затем добавляются компоненты. В качестве базовой ОС был выбран CentOS 7. В качестве FireWall выступает одно из устройств: ASA 5505 либо DFL-860E. На GW возложены не только функции шлюзов, но и прокси-сервера. Работ с VPN на Linux не проводится, т.к. есть лаборатории с полноценными аппаратными решениями. Выполнение таких работ возможно благодаря качественному оснащению лабораторий. УРПК им. А.С. Попова пытается стать образовательной площадкой, на базе которой студенты имеют возможность поработать на современном сетевом оборудовании (учебные классы таких вендоров как Cisco, D-Link, TP-Link), с современным программным и аппаратным обеспечением

Следует отметить, что учебные курсы производителей ориентированы в первую очередь на мир Windows, поэтому приходится прилагать титанические усилия для переработки учебных материалов на свободное программное обеспечение. Кроме этого, руководство УРПК прикладывает усилия по развитию ИТ инфраструктуры, с тем, чтобы она соответствовала современному состоянию информационных технологий в актуальном состоянии. Это требует систематического добавления новых интересных сервисов в процесс технического обслуживания образовательного процесса. СПО в этом неопределимо помогает, т.к. позволяет активно изучать интересные пакеты и аппаратным ресурсам, а самое главное, доступно для различных дополнений и модификаций под конкретные задачи.

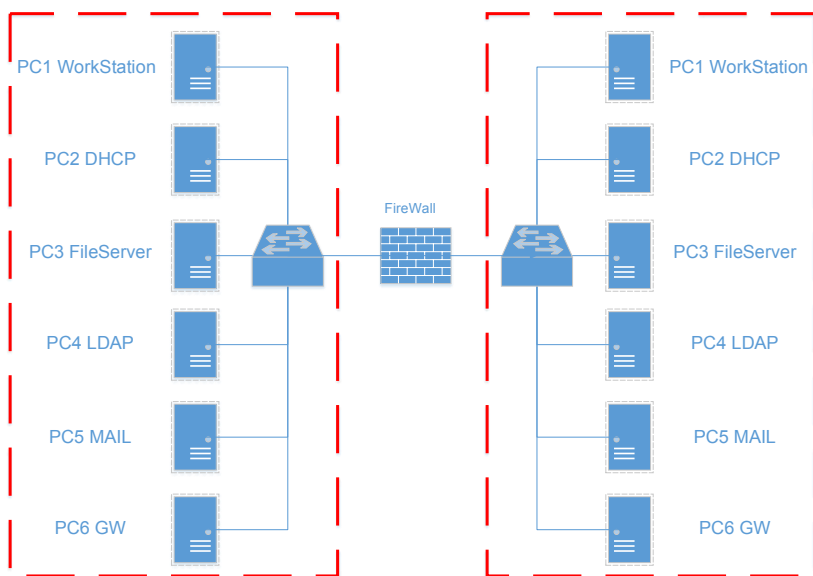


Рис. 1. Схема лабораторной работы

В настоящий момент для обслуживания лабораторий и сервисов тестируется система управления конфигурациями Ansible, которая подкупила относительной простотой при большой гибкости. Кроме этого, Ansible написан на Python, на Python пишутся и модули для него, а колледж активно пропагандируем этот язык программирования, и уже предлагает его изучение на нескольких специальностях в рамках дисциплины «Информатика».

В дальнейших планах переход на восьмую платформу Alt Linux, если они останутся бесплатными ;(и развёртывание LDAP-домена с целью централизованного управления правами студентов в лабораториях и обеспечения непрерывной рабочей среды. Помимо этого, мы начали сотрудничество с компанией РусБИТех, запланирована организация учебных мест на базе AstraLinuxSE.

1.Итак, свободное программное обеспечение позволяет Ит-специалистам более глубоко изучать технологии: если студент смог настроить DNS на Linux, то у него не возникнет проблем при аналогичных настройках на Windows или сетевом оборудовании.

2.Кризис и санкции стимулировали внедрение СПО в СрПО. Например, до 2015 года руководство особого интереса не проявляло, а в 2015 году был сделан о практике внедрения свободного программного

обеспечения в УРТК для Совета директоров ССУЗов Свердловской области.

3.Свободное программное обеспечение сложнее в развёртывании, настройке и обслуживании. Руководство не готово ни оплачивать полноценную техническую поддержку, ни доплачивать за обслуживание СПО, что влияет на мотивацию технического персонала.

4.Поскольку у Linux отсутствует маркетинговая поддержка, то затруднительно агитировать и продвигать свободные технологии в массы. Например, колледж проводит много различных мероприятий: областные и международные олимпиады. С одной стороны, поддержку данным мероприятиям готовы оказать только люди, интересующиеся СПО в частном порядке. Компании и вендоры не готовы оказывать поддержку. С другой стороны, большое количество учебных заведений испытывают трудности при подготовке студентов к заданиям с использованием СПО.

5.К сожалению, мы «нищоброды», которые пытаются готовить профессионалов, в чём СПО нам активно помогает.

Список использованных источников

1. Уймин А.Г., Ершова К.О. Перспективы развития специальностей информационно-технического профиля в рамках СПО [Текст] // Теория и практика науки третьего тысячелетия: сборник статей Международной научно-практической конференции. 7 марта 2014 г. / Под общ.ред. А.А. Суакисян. - Уфа: РИЦ БашГУ, 2014. - С. 275 – 279 - ISBN 978-5-7477-3501-9

2. Система управления Ansible [Электронный ресурс] / Режим доступа: <https://habrahabr.ru/company/selectel/blog/196620/>

Научное издание

ТЕОРИЯ И ПРАКТИКА ПРИМЕНЕНИЯ СВОБОДНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

*Сборник материалов Российской молодежной конференции
с элементами научной школы
18-22 мая 2016 г.*

Под общей редакцией
Гавриловой И.В., Назаровой О.Б.

Издается полностью в авторской редакции

Подписано в печать 09.06.2016. Рег. № 158-16. Формат 60×84/16. Бумага тип. № 1.
Плоская печать. Усл.печ.л. 14,00. Тираж 100 экз. Заказ 275.



Издательский центр ФГБОУ ВО «МГУ»
455000, Магнитогорск, пр. Ленина, 38
Полиграфический участок ФГБОУ ВО «МГУ»