

УДК 371.69

## ИНФОРМАЦИОННАЯ ИНФРАСТРУКТУРА ОБРАЗОВАТЕЛЬНОЙ СРЕДЫ С ИСПОЛЬЗОВАНИЕМ ТЕХНОЛОГИИ ВИРТУАЛЬНЫХ МАШИН

А.Д. Береснев<sup>a, b</sup>, Н.Ф. Гусарова<sup>a</sup>, Р.В. Иванов<sup>a</sup>, Д.Н. Федотов<sup>c, d</sup><sup>a</sup> Университет ИТМО, 197101, Санкт-Петербург, Россия<sup>b</sup> ООО «Ротэк Технолоджиз», 192007, Санкт-Петербург, Россия<sup>c</sup> Институт информатики и вычислительной техники (IATI AS), 10416, Таллин, Эстония<sup>d</sup> Учебный центр Omis OÜ, Satelles Saturni OÜ, 10117, Таллин, Эстония**Аннотация.**

**Предмет исследования.** Создана и исследована информационная инфраструктура среды обучения с применением технологии виртуальных машин для малых педагогических систем (отдельные классы, авторские курсы).

**Методика исследования.** В методологии ARIS построена модель жизненного цикла информационной инфраструктуры для малых педагогических систем с использованием виртуальных машин. Предложена методика формирования информационной инфраструктуры с виртуальными машинами на базе процессного подхода. В качестве базовой модели использована модель событийной цепочки в сочетании с диаграммой окружения. Для каждой функции событийной цепочки определен необходимый набор средств информационной и программной поддержки. Применение методики проиллюстрировано на примере проектирования информационной инфраструктуры образовательной среды, учитывающей специфику малых педагогических систем. К преимуществам спроектированной информационной инфраструктуры относятся: максимальное использование открытых или бесплатных компонентов; использование стандартных протоколов (в основном HTTP и HTTPS); максимальная переносимость (серверы приложений могут быть развернуты на любой из распространенных операционных систем); единый интерфейс к управлению различными платформами виртуализации; возможность инвентаризации содержимого виртуальной машины без ее запуска; гибкое управление инвентаризацией виртуальной машины с помощью настраиваемых цепочек правил.

**Апробация.** Апробация результатов работы проводилась на базе учебного центра «Институт информатики и вычислительной техники» (Таллин, Эстония). Применение методики в рамках курса «Пользование компьютером и программным обеспечением» позволило более чем в 2 раза сократить число отказов компонентов информационной инфраструктуры, требующих вмешательства технического специалиста, а также время на устранение такого рода неисправностей. Кроме того, учащиеся, получившие более широкий опыт работы с компьютером и программным обеспечением за счет применения информационной инфраструктуры с виртуальными машинами, показали более хорошие результаты в рамках зачетных испытаний.

**Практическое применение.** Результаты работы могут быть рекомендованы для внедрения в практику работы образовательных учреждений среднего и высшего образования, а также при разработке авторских и специализированных курсов, ориентированных на коллективную работу учащихся.

**Ключевые слова:** виртуальные машины, платформа виртуализации, процессный подход, малые педагогические системы, информационная инфраструктура, цепочки правил, событийная цепочка.

## INFORMATION INFRASTRUCTURE OF THE EDUCATIONAL ENVIRONMENT WITH VIRTUAL MACHINE TECHNOLOGY

A.D. Beresnev<sup>a, b</sup>, N.F. Gusarova<sup>a</sup>, R.V. Ivanov<sup>a</sup>, D.N. Fedotov<sup>c, d</sup><sup>a</sup> ITMO University, 197101, Saint Petersburg, Russia, natfed@list.ru;<sup>b</sup> "ROTEK Technologies", Ltd, 192007, Saint Petersburg, Russia, open.look@gmail.com<sup>c</sup> Institute of Informatics and Computers (IATI AS), 10416, Tallinn, Estonia<sup>d</sup> Education Center OMIS OÜ, Satelles Saturni OÜ, 10117, Tallinn, Estonia

**Subject of research.** Information infrastructure for the training environment with application of technology of virtual computers for small pedagogical systems (separate classes, author's courses) is created and investigated.

**Research technique.** The life cycle model of information infrastructure for small pedagogical systems with usage of virtual computers in ARIS methodology is constructed. The technique of information infrastructure formation with virtual computers on the basis of process approach is offered. The model of an event chain in combination with the environment chart is used as the basic model. For each function of the event chain the necessary set of means of information and program support is defined. Technique application is illustrated on the example of information infrastructure design for the educational environment taking into account specific character of small pedagogical systems.

Advantages of the designed information infrastructure are: the maximum usage of open or free components; the usage of standard protocols (mainly, HTTP and HTTPS); the maximum portability (application servers can be started up on any of widespread operating systems); uniform interface to management of various virtualization platforms, possibility of inventory of contents of the virtual computer without its start, flexible inventory management of the virtual computer by means of adjusted chains of rules.

**Approbation.** Approbation of obtained results was carried out on the basis of training center "Institute of Informatics and Computer Facilities" (Tallinn, Estonia). Technique application within the course "Computer and Software Usage" gave the possibility to get half as much the number of refusals for components of the information infrastructure demanding intervention of the technical specialist, and also the time for elimination of such malfunctions. Besides, the pupils who have got broader experience with computer and software, showed better results within tests due to the usage of information infrastructure with virtual computers.

**Practical application.** Obtained output can be recommended for introduction into the activity of educational institutions of secondary and higher education, and also for developing the author's and specialized courses focused on collective actions of pupils.

**Keywords:** virtual computers, virtualization platform, process approach, small pedagogical systems, information infrastructure, chains of rules, event chain.

## Введение

Реализация задач, поставленных Законом «Об образовании в Российской Федерации»<sup>1</sup> и государственными программами развития образования<sup>2,3</sup>, требует новых подходов к формированию информационной инфраструктуры (ИИ) в рамках единой образовательной среды<sup>4</sup>. Среди таких подходов важное место занимает создание сред обучения на базе технологии виртуальных машин (ВМ). ВМ определяется как «эффективный, изолированный дубликат реальной машины» [1] или как «программная реализация машины (т.е. компьютера), который выполняет программы как физическая машина» [2]. Содержательно под ВМ понимается вычислительная среда, реализованная в физическом компьютере и выделяющая его аппаратные ресурсы таким образом, чтобы предоставить возможность запуска нескольких операционных систем на одном компьютере. При этом каждая операционная система (ОС) выполняется в собственной ВМ и имеет выделенные логические экземпляры процессоров, жестких дисков, сетевых карт и других аппаратных ресурсов компьютера.

Виртуализация – одна из наиболее быстро развиваемых информационных технологий, которая активно внедряется и в сферу образования [3, 4]. Например, в [5] сообщается о разработке, основанной на виртуализации системы дистанционного обучения vMoodle, интегрированной с обучающей средой Moodle; vMoodle поддерживает ВМ, размещенные как в частных облаках, так и в общественном облаке (Amazon EC2). Несколько учебных курсов открытого университета Великобритании используют технологию ВМ [6]. Образовательные сети США [7] предоставляют ВМ как сетевой сервис. В практике работы образовательных учреждений России также имеется опыт использования ВМ (например, [8–11]).

Однако при попытке внедрения технологии виртуализации в практику своей работы педагог сталкивается с целым рядом проблем.

- В исследованиях и проектах последних лет широко представлены работы, посвященные оценке эффективности электронных образовательных ресурсов и информационных сервисов (см., например, [12–19]). Однако эти работы ориентированы на педагогические системы макро- и мезоуровней [20], т.е. на государственные и региональные системы образования, а также на учебные заведения в целом. Предлагаемые в них методики не позволяют учесть индивидуальный опыт и подготовленность педагога и обучаемых, существенные на микроуровне, т.е. в малых педагогических системах (МПС), например, в конкретном классе или авторском курсе, и не обеспечивают гибкой адаптации ИИ к конкретной педагогической задаче.
- Существуют различные технологии виртуализации, каждая из которых реализована в наборе коммерческих ВМ. Однако их сравнительные оценки, представленные в литературе, основываются на маркетинговых данных компаний-разработчиков (например, [21]) и не учитывают специфику работы ВМ в конкретном инфраструктурном решении, что затрудняет встраивание ВМ в систему средств обучения, уже используемую педагогом.

Таким образом, очевидные преимущества технологии ВМ в образовательном процессе реализуются только при наличии комплексного инфраструктурного решения, поддерживающего работу с ВМ в условиях конкретной педагогической системы. В то же время как показывает анализ, для МПС такие инфраструктурные решения отсутствуют. Целью настоящей работы является разработка методики формирования ИИ для использования технологии ВМ в условиях перехода к единой информационной образовательной среде. Применение методики иллюстрируется на примере проектирования ИИ образовательной среды, учитывающей специфику МПС.

### Методика формирования ИИ МПС с использованием технологии ВМ

ИИ как компонент архитектуры информационных технологий в общем случае является одним из механизмов реализации целей организации. В рассматриваемом случае в качестве основной цели выступает адекватное решение педагогической задачи, возникающей в рамках МПС, и ИИ с применением технологии ВМ правомерно рассматривать как педагогическое программное средство [20], обладающее специфическими особенностями (СО).

СО1. Компоненты ВМ, как правило, должны встраиваться в уже существующую информационную архитектуру МПС, причем потребность такого встраивания может возникнуть на любых этапах жизненного цикла ИИ.

---

<sup>1</sup> Федеральный закон от 29 декабря 2012 года №273-ФЗ «Об образовании в Российской Федерации»

<sup>2</sup> Государственная программа Российской Федерации «Развитие образования» на 2013–2020 годы, утверждена распоряжением Правительства РФ от 22.11.2012 № 2148-р

<sup>3</sup> Федеральная целевая программа развития образования на 2011–2015 годы, утверждена постановлением Правительства РФ от 7.2.2011 № 61

<sup>4</sup> Концепция развития единой информационной образовательной среды в Российской Федерации. – [http://raec.ru/upload/files/eios\\_conception.pdf](http://raec.ru/upload/files/eios_conception.pdf)

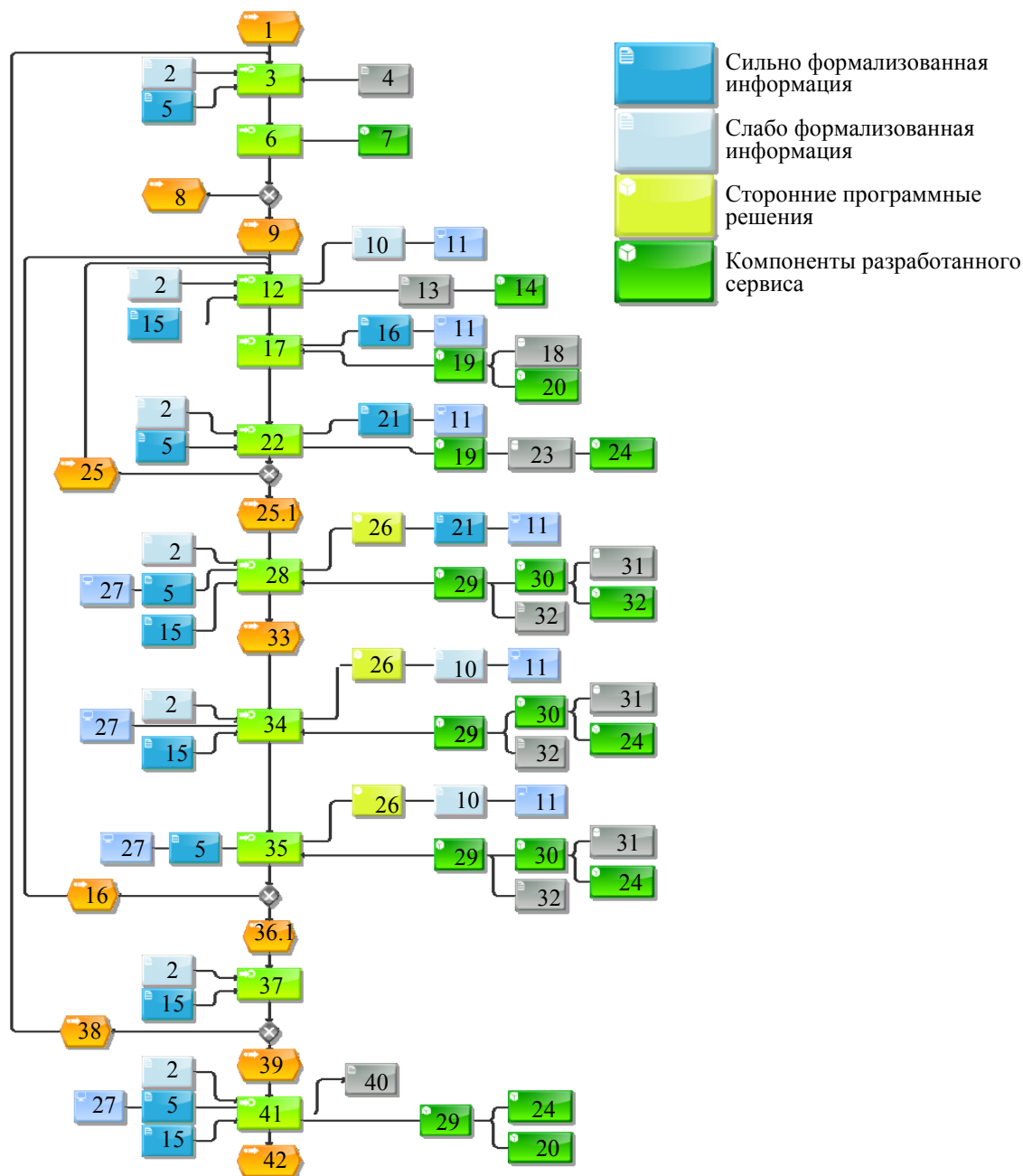


Рис. 1. Модель жизненного цикла ИИ МПС с использованием VM в методологии ARIS: 1 – старт педагогической задачи; 2 – информация о составе МПС; 3 – идентификация параметров педагогической задачи; 4 – шаблон параметров для оценки рисков; 5 – параметры имеющейся ИИ; 6 – оценка рисков применения ИИ; 7 – модуль оценки рисков; 8 – выбор ИИ без VM; 9 – переход к формированию ИИ с VM; 10 – сторонний опыт и примеры описания VM; 11 – среда Интернет; 12 – лингвистическое описание необходимой VM; 13 – структурированная информация о необходимой VM; 14 – тематический форум по VM; 15 – критерии достижения целей педагогической задачи; 16 – информация об имеющихся VM; 17 – отбор первичного набора возможных реализаций VM; 18 – уточнение технических параметров; 19 – система отбора VM; 20 – модуль семантического анализа; 21 – информация о параметрах имеющихся VM и (или) среды виртуализации; 22 – запуск VM; 23 – информация о технических параметрах выбранного прототипа VM; 24 – модуль сбора технических параметров; 25 – возможный вариант VM отсутствует; 25.1 – вариант VM выбран; 26 – среда виртуализации; 27 – характеристики существующей ИИ; 28 – тонкая настройка VM; 29 – модуль визуализации; 30 – варианты сред виртуализации; 31 – репозиторий VM; 32 – информация с форума; 33 – VM готова к эксплуатации в МПС; 34 – использование VM учащимися; 35 – контроль технических параметров ИИ с VM; 36 – технические параметры ИИ с VM не соответствуют педагогической задаче МПС; 36.1 – технические параметры ИИ с VM соответствуют педагогической задаче МПС; 37 – контроль соответствия параметров ИИ с VM критериям достижения цели педагогической задачи; 38 – педагогическая задача МПС не выполнена; 39 – педагогическая задача МПС выполнена; 40 – зафиксированные результаты выполнения педагогической задачи; 41 – завершение выполнения педагогической задачи; 42 – педагогическая задача завершена

СО2. Решение об использовании ВМ, как правило, принимается в рамках самой МПС, а не «спускается сверху» как обязательный компонент ИИ; целесообразно предусмотреть возможность гибкого пересмотра этого решения (вплоть до его отмены) при изменении ситуации в МПС.

СО3. При принятии решения об использовании ВМ необходимо уделять особое внимание человеческому фактору, т.е. индивидуальным особенностям каждого участника МПС.

СО4. ИИ с применением технологии ВМ является программным средством, и при оценке его качества необходимо использовать соответствующие ГОСТы<sup>1</sup>.

Как показал сравнительный анализ, наиболее полно учесть эту специфику позволяет процессный подход к моделированию ИИ, реализуемый в методологии ARIS [22]. Эта методология дает возможность комплексно описать ИИ не только как сугубо программную сущность, но и как компонент социально-технической системы. В качестве базовой модели для формирования ИИ МПС выбрана модель событийной цепочки (Extended event driven process chain, eEPC), которая описывает последовательность управляющих событий и функций, выполняемых ИИ в ходе решения педагогической задачи.

Существенным преимуществом методологии ARIS является возможность комбинирования модели eEPC с диаграммой окружения (Function allocation diagram), что позволяет сопоставить каждой функции модели событийной цепочки необходимый набор средств информационной и программной поддержки. На рис. 1 представлена объединенная модель в нотации ARIS, причем, в дополнение к типовым графическим обозначениям элементов нотации ARIS, для средств информационной и программной поддержки применен цветовой код (см. врезку). Модель отражает основные этапы жизненного цикла использования ВМ в ИИ МПС, содержательно соответствующие следующим вопросам:

1. Вводить ли ВМ в состав ИИ для решения педагогической задачи?
2. Какой тип ВМ использовать?
3. Не требуется ли текущая модификация ВМ в составе ИИ?

Первый этап (блоки 1, 3, 6, 8, 9 событийной цепочки) предваряет переход к ИИ с применением ВМ, поэтому здесь производится проактивная оценка рисков такого перехода для решения заданной педагогической задачи в условиях конкретной МПС и существующего состоянии ИИ. Для поддержки этого этапа авторами разработан проблемно-ориентированный инструментарий оценки рисков [23].

На втором этапе (блоки 12, 17, 22, 25, 25.1 событийной цепочки) выполняется выбор типа ВМ для построения ИИ как компонента образовательной среды. Критерии этого выбора определены вышеупомянутыми стандартами, однако, как показывает анализ, для решения большинства реальных педагогических задач многие из этих критериев оказываются взаимосвязанными. В связи с этим здесь необходим инструментарий многокритериального анализа, позволяющий выявить взаимосвязи между конкретными критериями и сократить их количество, представленный в работе авторов [24]. Этап завершается «тонкой настройкой», т.е. кастомизацией выбранного прототипа в соответствии с параметрами уже существующих компонентов ИИ, куда встраивается ВМ, а также спецификой МПС (блоки 28, 33 событийной цепочки).

После введения настроенной ВМ в состав ИИ начинается решение педагогической задачи. На этом этапе осуществляется постоянный мониторинг адекватности текущего состояния структуры и параметров ИИ решаемой педагогической задаче и, при необходимости, модификация структуры и параметров ИИ вплоть до возвращения к структуре ИИ без ВМ (блоки 34–39 событийной цепочки). Здесь требуется программный инструментарий для поддержки мониторинга. Отдельные программные решения для такого мониторинга, разработанные авторами, защищены свидетельствами на интеллектуальную собственность [25–27]. Этап завершается фиксацией результатов деятельности МПС (блоки 41, 42 событийной цепочки). Проведенный анализ событийной цепочки жизненного цикла ИИ в сочетании с его особенностями (СО1–СО4) позволяет для каждой из выделенных функций модели выделить необходимые средства информационной и программной поддержки, которые также представлены на рис. 1 (расшифровка – в подрисуночной подписи). Заметим, что многие из этих средств для различных функций дублируют друг друга, что упрощает построение ИИ в целом; с другой стороны, особенности СО1–СО4 вынуждают учитывать при выборе программных средств для каждой функции возможный уровень формализации ее информационной поддержки, что отражено в легенде рис. 1.

Таким образом, методика формирования ИИ МПС с использованием технологии ВМ может быть представлена в виде следующей последовательности шагов:

1. идентифицировать педагогическую задачу;
2. определить параметры списка СО1–СО4 применительно к конкретной МПС;
3. представить функции, выполняемые ИИ под управлением педагога в ходе решения педагогической задачи, в виде событийной цепочки;

<sup>1</sup> ГОСТ 25195-89 «Оценка качества программных средств. Общие положения»; ГОСТ Р ИСО/МЭК 9126 «Информационная технология. Оценка программной продукции. Характеристика качества и руководства по их применению»; ГОСТ 28806-90. Качество программных средств. Термины и определения

4. используя результаты п.п. 1 и 2, для каждой функции событийной цепочки определить необходимый набор средств информационной и программной поддержки (объектов окружения);
5. в результатах п. 4 устранить дублирование;
6. используя результаты п.п. 2 и 5, перейти к проектированию ИИ МПС.

### Архитектура ИИ МПС с использованием технологии VM

Содержательный анализ и факторизация объектов окружения событийной цепочки (рис. 1) позволяют перейти к проектированию ИИ МПС с использованием технологии VM.

На рис. 2 приведена архитектура ИИ в нотации UML в виде диаграммы развертывания. Для раскрытия основных принципов функционирования ИИ диаграмма дополнена компонентами системы. ИС построена по трехзвенной архитектуре с разнесением слоя сервера приложений на два узла. К основным задачам ИС относятся:

- сбор формальных данных о VM;
- сбор слабо формализуемой информации о VM;
- подготовка сводных отчетов о свойствах VM;
- единообразное управление VM на разных гипервизорах.

В связи с выделенными выше особенностями МПС (CO1 и CO3) в архитектуре предусматривается возможность работы VM на гипервизорах, развернутых на отдельных устройствах, причем с одновременной поддержкой нескольких платформ. В данной реализации предусмотрена поддержка трех платформ: Microsoft Hyper-V, VMware vSphere и Proxmox Virtual Environment, каждая из которых может быть представлена несколькими экземплярами. На каждой платформе может выполняться множество VM. На схеме они сгруппированы в пакет «Платформы виртуализации».

Архитектурным ограничением является использование в VM исключительно эмулируемых дисковых устройств (образов виртуальных жестких дисков), даже в том случае, если гипервизор поддерживает прямой доступ к дисковым устройствам. Хранение образов VM и их файлов конфигурации организовано в единой сетевой системе хранения данных (NAS). Доступ к файлам VM с платформ виртуализации осуществляется по протоколу SMB или NFS. Единая сетевая система хранения данных вместо сети хранения данных выбрана для упрощения совместного доступа к файлам образов VM и конфигураций со стороны гипервизоров и подсистемы инвентаризации VM.

Клиентская часть построена в виде Web-приложения с ограничением браузеров по реализации на актуальных программных ядрах Trident, WebKit и Gecko, поддерживающих HTML5. Клиентская часть не требует установки специальных плагинов и не предполагает выполнение апплетов Java или flash-программ. Web-клиент, выполняя стандартную задачу интерфейсного ввода-вывода, взаимодействует с Web-сервером, на котором развернуты Web-модули сайта front-end (пакет «Web-модули front-end»).

На сервере приложений подсистемы поддержки принятия решения (ППР) функционирует сервер приложений Tomcat, выполняющий сервлеты (пакет «Web-модули back-end»). На этом сервере работает подсистема инвентаризации VM, осуществляющая управляемый сбор формальных данных о конфигурациях VM, о конфигурации операционных систем VM и об установленном в VM программном обеспечении (ПО). Для реализации прозрачного и единообразного управления VM, выполняемыми на разных гипервизорах, в системе предусмотрен промежуточный сервер приложений подсистемы управления VM. На нем развернут управляющий модуль, который через стандартные компоненты обеспечивает управление VM, а также доступ к консолям VM прямо в окне браузера. На сервере баз данных (БД) развернуты экземпляры MS SQL Server и MS SQL Server Analysis Services. Система управления базами данных обеспечивает работу трех баз данных, показанных на рис. 2. Состав пакетов «Web-модули front-end» и «Web-модули back-end» показан на рис. 3.

Модуль пользовательского интерфейса является точкой входа клиента на сайт. Через него подключаются внешние модули. Модуль управления реализует пользовательский интерфейс к управлению VM. Команды пользователя об изменении состояния VM (создание, клонирование, остановка, запуск, редактирование свойств и т.п.) передаются к подсистеме управления VM, которая, в свою очередь, или через Python-скрипты по протоколу SSH или через скрипты PowerShell по протоколу RPC/HTTPS передает их гипервизорам. Также модуль обеспечивает доступ к консолям VM в окне браузера через Proxu-модуль подсистемы управления VM.

Модуль форума jforum обеспечивает коммуникацию пользователей в режиме отложенного времени. Следует отметить, что, в соответствии с выявленным содержанием объектов окружения в составе ИИ (рис. 1), основное назначение форума – сбор слабо формализованной информации о VM. В сообщениях форума каждый экземпляр VM идентифицируется уникальным составным идентификатором, состоящим из имени сервера виртуализации и ID VM.

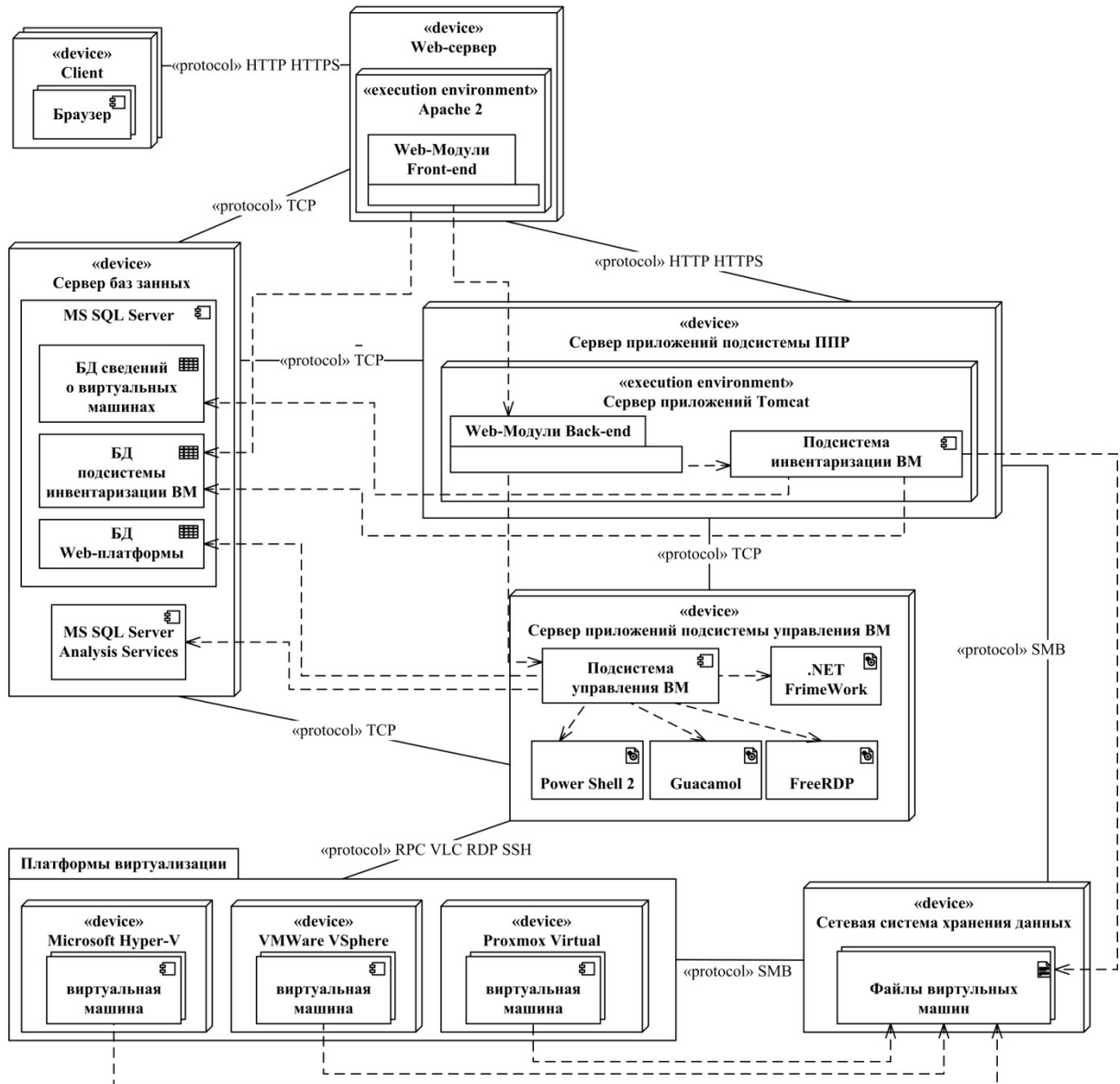


Рис. 2. Диаграмма развертывания ИИ МПС с использованием ВМ

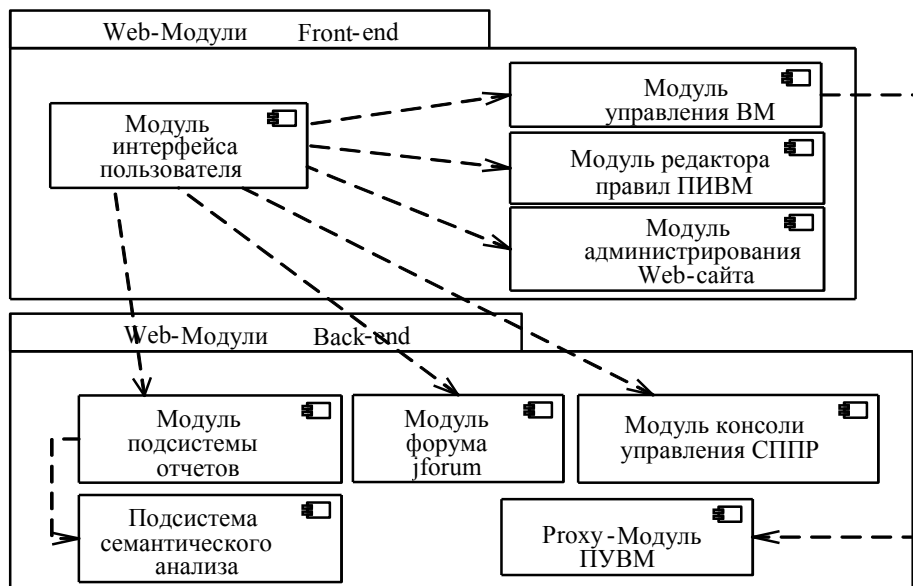


Рис. 3. Диаграмма компонентов пакетов Web-сайтов

В подсистему поддержки принятия решения объединены:

- модуль подсистемы отчетов, отвечающий за передачу запросов и за формирование сводных отчетов;
- подсистема семантического анализа, обеспечивающая формирование запросов и доступ к результатам обработки системой MS SQL Server Analysis Services данных о ВМ;
- модуль консоли управления, предназначенного для управления шаблонами отчетов.

### Специфика реализации ИИ МПС с использованием ВМ

Как подчеркивалось выше, в связи с особенностями МПС к архитектуре и программной реализации ИИ с ВМ предъявляются специфические требования:

- в ИИ могут включаться ВМ на гипервизорах, развернутых на отдельных устройствах, причем с одновременной поддержкой нескольких платформ; педагог-администратор должен иметь возможность контролировать текущий состав ИИ без запуска ВМ;
- педагог-администратор ИИ должен иметь возможность контролировать текущее состояние каждой запущенной ВМ и при необходимости удаленно осуществлять необходимые корректирующие действия.

Ключевыми компонентами ИИ для реализации этих требований являются подсистема инвентаризации ВМ и подсистема удаленного управления.

**Подсистема инвентаризации ВМ** предназначена для сбора формальных данных о виртуальных машинах. К этим данным относятся:

- составное наименование ВМ;
- информация о платформе виртуализации (модель, версия);
- информация о конфигурации виртуального оборудования (количество ядер CPU, объем, количество и объем жестких дисков, интерфейс подключения жестких дисков, количество и скорость работы сетевых адаптеров);
- информация о средствах интеграции (наличие, версия, поддержка управления питанием, поддержка heartbeat);
- версия виртуальной ОС;
- информация об установленном в ВМ программном обеспечении и, при необходимости, параметрах его конфигурации.

Особенностью функционирования подсистемы является отсутствие необходимости запуска ВМ для получения необходимой информации. Исходными объектами для нее служат файлы конфигурации ВМ и файл виртуального диска. Функционирование программы не зависит от типа ОС, установленной в ВМ.

Подсистема инвентаризации ВМ осуществляет анализ ВМ, выполняя набор специальных управляющих директив (правил). Правила представляют собой функции, получающие в качестве аргумента данные об объекте анализа и возвращающие строковые данные и (или) вызывающие следующие правила. Структурно правило представляет собой совокупность обработчика и дополнительных данных для обработчика. Обработчик – это шаблонная функция, имеющая одинаковый формат входных и выходных данных. Это и позволяет составлять из них цепочки, работающие по принципу конвейера – выход одной функции является входом другой. В описании обработчика содержатся типы данных, которые требуются на входе, и тип данных, получаемых на выходе. Это описание используется при проверке цепочки правил при добавлении в базу и выполнении. В случае нарушения осуществляется переход к следующей цепочке и выдается сообщение об ошибке. Кроме данных, полученных из предыдущего правила, обработчик принимает дополнительные данные, содержащиеся в описании правила, или в описании цепочки.

Эксперт заранее готовит упорядоченные наборы (цепочки) правил, определяя объекты для анализа, конкретные параметры ВМ, значения которых следует определить, а также последовательность проверки и условия перехода между цепочками. Для этого служит модуль редактора правил.

Цепочки правил хранятся в специальной базе данных. В программе цепочка правил представлена стеком правил. Обработка цепочки правил реализована в виде рекурсивного вызова обработки правил. В качестве входных данных в следующее правило передается результат работы текущего правила (для первого элемента передается NULL) и стек цепочки правил, из которого текущее правило удаляет свою запись.

Таким образом, подсистема инвентаризации по файлам конфигурации ВМ на сетевом хранилище определяет параметры ВМ и по содержимому виртуальных дисков проводит инвентаризацию установленного в ВМ программного обеспечения и его конфигурации. В этом случае объектом анализа выступают пары «файл конфигурации с регулярной структурой – параметр в этом файле» или «файл бинарных данных с регулярной структурой – указатель на данные». В настоящее время реализован анализ файлов ini, conf (linux) и реестра MS Windows. Запуск проверки осуществляется или по выполнению триггера, или автоматически при фиксации релиза ВМ, или вручную для любого экземпляра ВМ, даже без фиксации релиза. На рис. 4 представлена sequence-диаграмма процесса сбора данных о ПО, развернутом в ВМ.

Дополнением к подсистеме инвентаризации ВМ может служить подсистема семантического анализа, которая обрабатывает тексты обсуждений ВМ на форуме, выделяя дополнительную информацию о ВМ.

**Подсистема удаленного управления ВМ** предназначена для управления различными виртуальными машинами, работающими на различных платформах виртуализации (вне зависимости от конкретной платформы виртуализации), при помощи Web-браузера. Она представляет собой Web-сервис, позволяющий управлять ВМ в сети, и реализует единую точку управления набором серверов виртуализации, в том числе:

- предоставление интерфейса регистрации серверов виртуализации;
- управление иерархией ВМ;
- управление отдельной ВМ (модификация, запуск, перезагрузка и т.п.);
- создание новой ВМ с управлением параметрами машины;
- работа с ВМ через консоль в окне браузера.

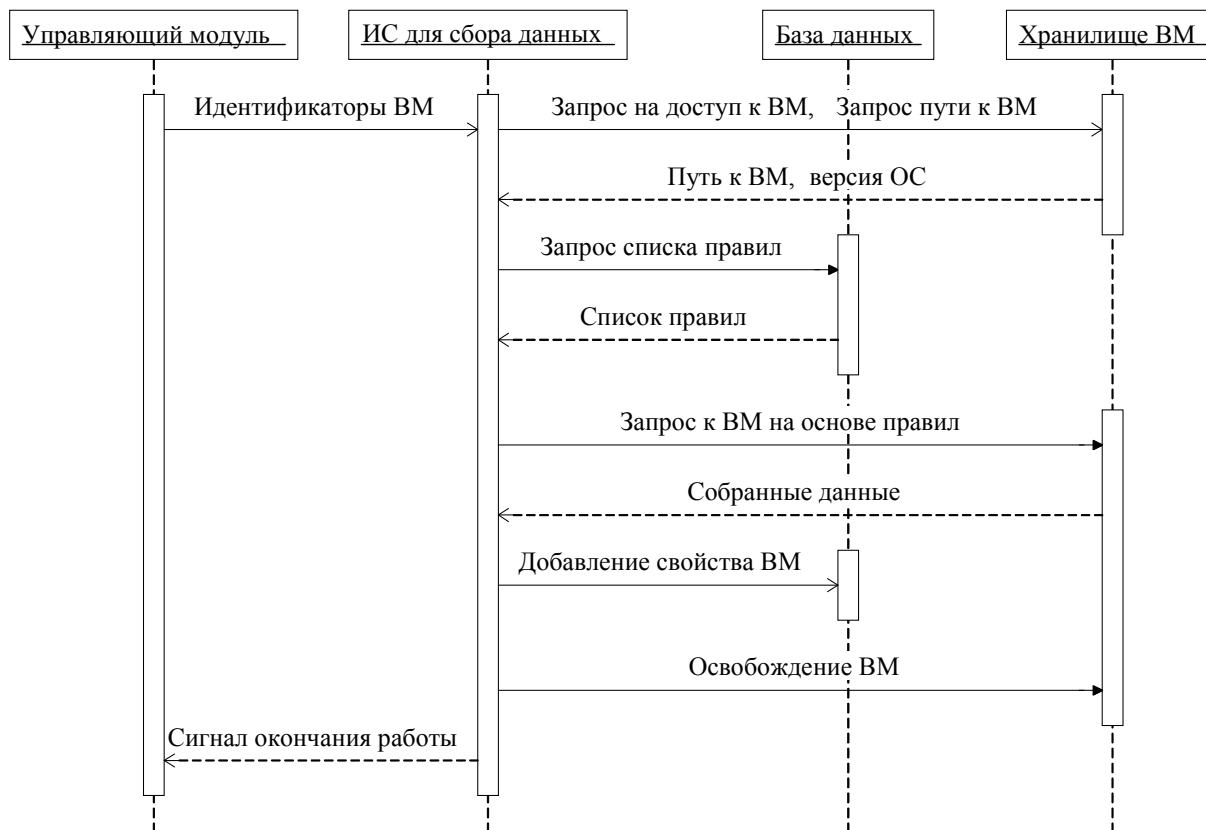


Рис. 4. Диаграмма процесса сбора данных

В подсистеме осуществляется поддержка трех платформ виртуализации: Microsoft Hyper-V, VMware vSphere и Proxmox Virtual Environment. Каждая платформа в системе может быть представлена несколькими экземплярами. Отметим, что номенклатура поддерживаемых платформ может быть расширена за счет незначительной модификации подсистемы.

Подсистема выполнена в виде модуля, реализованного с использованием технологий Microsoft .NET на языке C#. Взаимодействие с гипервизорами осуществляется через стандартные для платформ виртуализации командные интерфейсы. Задача модуля сводится к установлению защищенного сеанса связи с платформой виртуализации и передачи высокоуровневых команд с интерпретацией ответов систем для их представления в едином интерфейсе пользователя.

Для платформы MS Hyper-V связующим элементом выбран Windows PowerShell – расширяемое средство автоматизации, состоящее из оболочки с интерфейсом командной строки и сопутствующего языка сценариев. Защищенный канал формируется с помощью службы WinRM, запущенной на стороне платформы. Трафик инкапсулируется в протокол HTTPS. Для остальных платформ защищенный канал формируется по протоколу SSH с взаимной аутентификацией по сертификатам.

Для доступа к консолям операционных систем ВМ используется схема на основе свободно доступных компонентов. Поток консольного ввода–вывода через стандартные для платформ виртуализации протоколы (VNC для VMware vSphere и Proxmox Virtual Environment, и RDP для Microsoft Hyper-V) поступает через открытые компоненты Guacamole и FreeRDP на Proxy-модуль, реализованный в виде JAVA сервлета; тот, в свою очередь, транслирует их в поток протокола HTML5 и передает их в модуль управления ВМ. Обратная связь реализуется аналогично.



### Апробация

Апробация результатов работы проводилась на базе учебного центра «Институт информатики и вычислительной техники» (Таллинн, Эстония). Для апробации была выбрана дисциплина «Пользование компьютером и программным обеспечением», входящая в базовый цикл. Длительность изучения дисциплины – 1 семестр. Занятия проводятся в группах по 12–15 человек. В рамках дисциплины учащиеся знакомятся с физическим и логическим устройством компьютера, обучаются пользованию основным набором прикладных программных продуктов. Использование ВМ в рамках данной дисциплины было направлено на то, чтобы учащиеся познакомились не только с нативным ПО фирмы Apple, которое является базовым для Центра, но и с ПО, разработанным под Windows и Linux. Выбор типа использованной ИИ, а также мониторинг деятельности учащихся выполнялись на основе разработанной методики.

Для сравнения было взято два потока: в одном из них проводились занятия с использованием ВМ, в другом – без использования ВМ. Сравнение проводилось по следующим характеристикам.

1. Количество отказов ПО, с использованием которого проводится занятие – отказов типа 1 (фиксировалось по обращениям учащегося к преподавателю, рассматривались только те отказы, для которых было необходимо вмешательство технического специалиста).
2. Количество отказов аппаратного обеспечения и ОС – отказов типа 2 (фиксировалось по обращениям преподавателя к техническому специалисту).
3. Среднее время восстановления работоспособности (оценивалось время, потраченное на устранение неисправности типов 1 и 2).
4. Личный опыт участников (оценивалось количество и сложность заданий, выполненных на зачетном занятии. Оценка приводится в 10 бальной шкале – средняя по группе).

Результаты апробации приведены в таблице.

Характеристика	Группа с ВМ	Группа без ВМ
Общее количество отказов ПО, ед.	3	7
Количество отказов техники, ед.	2	2
Среднее время восстановления (тип 1/тип 2), ч	0,5/2	1/2
Личный опыт участников, баллов (по 10 бальной шкале)	9	7

Таблица. Результаты апробации

Полученные результаты подтверждают эффективность предложенной методики создания ИИ МПС на базе ВМ: более чем в 2 раза сократилось число отказов компонентов ИИ, требующих вмешательства технического специалиста, а также снизилось время на устранение такого рода неисправностей. Кроме того, учащиеся, получившие более широкий опыт работы с компьютером и ПО за счет применения ИИ с ВМ показали более хорошие результаты в рамках зачетных испытаний.

### Заключение

В работе предложена методика формирования информационной инфраструктуры для использования технологии виртуальных машин в условиях перехода к единой информационной образовательной среде. Применение методики проиллюстрировано на примере проектирования информационной инфраструктуры образовательной среды, учитывающей специфику малой педагогической системы. К особенностям спроектированной информационной инфраструктуры относятся:

- максимальное использование открытых или бесплатных компонентов. Так, в качестве сред выполнения используются серверы Apache и Tomcat (лицензия Apache), модуль Guacamole (лицензии AGPL 3.0), модуль FreeRDP (лицензия Apache), PowerShell (бесплатный в рамках EULA);
- использование стандартных протоколов (в основном HTTP и HTTPS);
- максимальная переносимость (серверы приложений могут быть развернуты на любой из распространенных операционных систем);
- единый интерфейс к управлению различными платформами виртуализации;
- возможность инвентаризации содержимого виртуальной машины без ее запуска;
- гибкое управление инвентаризацией виртуальной машины с помощью настраиваемых цепочек правил.

### Литература

1. Popek G.J., Goldberg R.P. Formal requirements for virtualizable third generation architectures // Communications of the ACM. 1974. V. 17. N 7. P. 412–421.
2. Smith J.E., Nair R. Virtual Machines: Versatile Platforms for Systems and Processes. Morgan Kaufmann, 2005. 656 p.
3. Гультьев А.К. Виртуальные машины: несколько компьютеров в одном. СПб: Питер, 2006. 224 с.
4. Блинков Ю.В. Моделирование компьютерных систем на виртуальных машинах. Пенза: ПГУАС, 2011. 268 с.

5. vMoodle Virtualization-Based Online Learning System [Электронный ресурс]. Режим доступа: <http://visa.cis.fiu.edu/tiki/vMoodle>, свободный. Яз. англ. (дата обращения 15.07.2014).
6. Cursory Thoughts on Virtual Machines in Distance Education Courses [Электронный ресурс]. Режим доступа: <http://blog.ouseful.info/2013/12/02/packaging-software-for-distance-learners-vms-101>, свободный. Яз. англ. (дата обращения 15.07.2014).
7. Virtual Machines - Kansas Research and Education Network [Электронный ресурс]. Режим доступа: <http://www.kanren.net/index.php/services/net-services/kanrenvm.html>, свободный. Яз. англ. (дата обращения 15.07.2014).
8. Скушдлевский А.А., Лямин А.В. Разработка стенда для экспериментальных исследований типовых схем развертывания системы AcademicNT // Научно-технический вестник СПбГУ ИТМО. 2011. № 3 (73). С. 126–129.
9. Усманов Ш.Н. Виртуальные машины в преподавании информатики // Информатика и образование. 2007. № 6. С. 120–121.
10. Чурилов И.А. Применение виртуальных машин в процессе обучения ИТ-специальностям // Преподавание информационных технологий в Российской Федерации: материалы XI открытой Всероссийской конференции. Воронеж: Воронежский государственный университет, 2013. С. 75–77.
11. Лисовецкий К. Виртуальная Windows в системе Linux [Электронный ресурс]. Режим доступа: [http://gimnaz.com/blog/win\\_to\\_linux.90.aspx](http://gimnaz.com/blog/win_to_linux.90.aspx), свободный. Яз. рус. (дата обращения 16.02.2014).
12. Гаврилов С.И. Модели, методы и программные средства оценки качества информационно-образовательных ресурсов: автореф. дис. ... канд. техн. наук. М.: МИФИ, 2011. 22 с.
13. Развитие электронных образовательных интернет-ресурсов нового поколения [Электронный ресурс]. Режим доступа: <http://kkidppo.ru/razvitie-eog-novogo-pokoleniya>, свободный. Яз. рус. (дата обращения 30.08.2014).
14. Ившина Г.В. Разработка электронных образовательных ресурсов: мониторинг качества и внедрение. Казань, КГУ, 2008. 53 с.
15. Структура системы оценки качества программных комплексов для дистанционного обучения [Электронный ресурс]. Режим доступа: <http://www.hr-portal.ru/article/struktura-sistemy-ocenki-kachestva-programmnyh-kompleksov-dlya-distancionnogo-obucheniya>, свободный. Яз. рус. (дата обращения 15.03.2014).
16. Журавлёва Е.В. Проблема качества педагогических программных средств в контексте инноватизации преподавания математики // Вестник РУДН. Серия: Информатизация образования. 2011. № 1. С. 85–90.
17. Бабешко В.Н., Нежурина М.И. Комплексные решения ЦДО МИЭМ в области качества электронного обучения [Электронный ресурс]. Режим доступа: <http://tm.ifmo.ru/tm2005/src/264b.pdf>, свободный. Яз. рус. (дата обращения 15.03.2014).
18. Wu H.-Y., Lin Y.-K., Chang C.-H. Performance evaluation of extension education centers in universities based on the balanced scorecard // Evaluation and Program Planning. 2011. V. 34. N 1. P. 37–50.
19. Tzeng G.H., Chiang C.H., Li C.W. Evaluating intertwined effects in e-learning programs: a novel hybrid MCDM model based on factor analysis and DEMATEL // Expert Systems with Applications. 2007. V. 32. N 4. P. 1028–1044.
20. Безрукова В.С. Основы духовной культуры (энциклопедический словарь педагога). Екатеринбург, 2000. 937 с.
21. Сравнение виртуальных машин [Электронный ресурс]. Режим доступа: [http://ru.wikipedia.org/wiki/Сравнение\\_виртуальных\\_машин](http://ru.wikipedia.org/wiki/Сравнение_виртуальных_машин), свободный. Яз. рус. (дата обращения 15.03.2014).
22. Шеер А.В. ARIS – моделирование бизнес-процессов. Вильямс, 2009. 175 с.
23. Артемова Г.О., Гусарова Н.Ф., Иванов Р.В. Оценка рисков применения виртуальных машин в инфраструктуре образовательного учреждения // Дистанционное и виртуальное обучение. 2014. № 11. (принято к печати).
24. Артемова Г.О., Гусарова Н.Ф., Иванов Р.В. Управление информационной инфраструктурой образовательной среды с использованием технологии виртуальных машин // Дистанционное и виртуальное обучение. 2014. № 10. (принято к печати).
25. Гусарова Н.Ф., Добренко Н.В., Иванов Р.В., Решетников В.В. Среда управления виртуальными машинами. Свидетельство о государственной регистрации программы для ЭВМ 2013661663. Заявл. 29.10.2013.
26. Гусарова Н.Ф., Добренко Н.В., Иванов Р.В., Прокофьев Е.С. Утилита извлечения структурированной информации о программном обеспечении. Свидетельство о государственной регистрации программы для ЭВМ 2013661662. Заявл. 29.10.2013.
27. Гусарова Н.Ф., Добренко Н.В., Иванов Р.В., Строганов П.С. Программа сопровождения поиска виртуальных машин в репозитории. Свидетельство о государственной регистрации программы для ЭВМ 2013661664. Заявл. 29.10.2013.

- Береснев Артем Дмитриевич*** – старший преподаватель, Университет ИТМО, 197101, Санкт-Петербург, Россия; руководитель ИТ-отдела, ООО «РОТЭК Технолоджиз», 192007, Россия, Санкт-Петербург, open.look@gmail.com
- Гусарова Наталия Федоровна*** – кандидат технических наук, старший научный сотрудник, доцент, Университет ИТМО, 197101, Санкт-Петербург, Россия, natfed@list.ru
- Иванов Роман Владимирович*** – старший преподаватель, Университет ИТМО, 197101, Санкт-Петербург, Россия, старший преподаватель, rvivanov@mail.ifmo.ru
- Федотов Дмитрий Николаевич*** – ведущий преподаватель, Институт информатики и вычислительной техники (ИАТИ АС), 10416 Таллинн, Эстония; ведущий преподаватель, Учебный центр Omis OÜ, Satelles Saturni OÜ (отдел информационных и методических технологий), 10117, Таллинн, Эстония, dmitry@satellessaturni.eu
- Artem D. Beresnev*** – lecturer, ITMO University, 197101, Saint Petersburg, Russia; Head of IT-department, “ROTEK Technologies”, Ltd, 192007, Saint Petersburg, Russia, open.look@gmail.com
- Natalia F. Gusarova*** – PhD, senior scientific researcher, Associate professor, ITMO University, 197101, Saint Petersburg, Russia, natfed@list.ru
- Roman V. Ivanov*** – senior lecturer, ITMO University, 197101, Saint Petersburg, Russia, rvivanov@mail.ifmo.ru
- Dmitry N. Fedotov*** – – leading lecturer, Institute of Informatics and Computers (IATI AS), 10416, Tallinn, Estonia; Education Center OMIS OÜ, Satelles Saturni OÜ, 10117, Tallinn, Estonia; dmitry@satellessaturni.eu

*Принято к печати 30.07.14*

*Accepted 30.07.14*