

Разработка прототипа многоагентной системы сетевого взаимодействия учебных заведений*

Целью исследования является разработка прототипа многоагентной интеллектуальной системы динамического взаимодействия интеллектуальных агентов в интегрированном информационно-образовательном пространстве для решения задачи формирования совместных образовательных программ несколькими учебными заведениями.

Материалы и методы. В современных условиях цифровой трансформации образования все более актуальное значение приобретает организация сетевого обучения студентов по динамически формируемым образовательным программам в соответствии с потребностями рынка труда и индивидуальными потребностями обучающихся. Для гибкой интеграции образовательных ресурсов и реализации совместных образовательных программ несколькими взаимодействующими учебными заведениями предлагается разработка программной платформы на базе интеллектуальной многоагентной технологии. В качестве основы для разработки архитектуры программного прототипа выбраны спецификации сообщества разработчиков по стандартизации агентных технологий FIPA (The Foundation for Intelligent Physical Agents), а программной инструментальной среды — фреймворк JADE (Java Agent Development Network).

Результаты. В статье представлена архитектура интеллектуальной многоагентной системы сетевого взаимодействия учебных заведений в интегрированном информационно-образовательном пространстве, позволяющая динамически формировать

образовательные программы в соответствии с запрашиваемыми профессиональными компетенциями. Обоснована структура онтологии информационно-образовательного пространства, обеспечивающая взаимодействие интеллектуальных агентов, и описан механизм ее отображения из формата OWL в формат инструментальной среды JADE с использованием плагина Protege. Представлено описание программного прототипа, структуры интеллектуальных агентов в формате JADE и технологии взаимодействия агентов на основе протоколов FIPA в процессе формирования образовательных программ.

Заключение. Реализация программного прототипа многоагентной системы сетевого взаимодействия учебных заведений позволяет оперативно создавать образовательные программы в соответствии с индивидуальными и групповыми траекториями обучения под конкретно формируемые профессиональные компетенции. Представленный программный прототип при некоторой модификации может использоваться для других предметных областей цифровой экономики, предполагающих динамическое формирование сетевых структур взаимодействия партнеров по бизнесу.

Ключевые слова: информационно-образовательное пространство, образовательные программы, сетевое взаимодействие, многоагентная система, онтология, протоколы взаимодействия агентов

Yury F. Telnov¹, Andrey V. Danilov¹, Rustem I. Diveev², Vasily A. Kazakov¹, Elena V. Yaroshenko¹

¹Plekhanov Russian University of Economics, Moscow, Russia

²Glowbyte Consulting, Moscow, Russia

Development of a prototype of multi-agent system of network interaction of educational institutions

The aim of the research is to develop a prototype of the intelligent multi-agent system for dynamic interaction of the intelligent agents in the integrated information and educational space to solve the problem of formation of joint educational programs by several educational institutions.

Materials and methods. In modern conditions of digital transformation of education the organization of network training of students on dynamically formed educational programs in accordance with the needs of the labor market and the individual requirements of students is becoming increasingly important. It is proposed to develop a software platform based on intelligent multi-agent technology for flexible integration of educational resources and implementation of joint educational programs by several interacting educational institutions. As a basis for the development of the software prototype architecture, the specifications of the developer community for the standardization of agent technologies FIPA (the Foundation for Intelligent Physical Agents), and the software tool environment — JADE framework (Java Agent Development Network) were chosen.

Results. The paper presents the architecture of intelligent multi-agent system for network interaction of educational institutions in the integrated information and educational space, which allows to dynami-

cally forming educational programs in accordance with the requested professional competencies. The structure of the ontology of information and educational space, providing the interaction of intelligent agents, is justified, and the mechanism of its display from the OWL format to the format of the tool environment JADE, using the plugin Protege is described. The description of the software prototype, the structure of intelligent agents in the JADE format and the technology of agent interaction, based on the FIPA protocols in the process of educational programs formation is presented.

Conclusion. The implementation of the multi-agent system prototype for network interaction of educational institutions allows you to quickly create educational programs in accordance with individual and group learning trajectories under the specific formed professional competence. The presented software prototype with some modification can be used for other subject areas of the digital economy, involving the dynamic formation of network structures of interaction for business partners.

Keywords: information and educational space, educational programs, network interaction, multi-agent system, ontology, agent interaction protocols

*Работа выполнена при поддержке РФФИ, гранты № 16-07-01062, 18-07-00918

Введение

В соответствии с Федеральным законом «Об образовании в Российской Федерации» [1] сетевая форма реализации образовательных программ обеспечивает возможность освоения обучающимся образовательной программы с использованием ресурсов нескольких организаций, осуществляющими образовательную деятельность. Реализация таких форм сетевых взаимодействия немыслима без создания интегрированного информационно-образовательного пространства (ИОП) на основе современных цифровых технологий, представляющего по сути распределенную программно-техническую образовательную платформу [2].

Проблеме создания и использования информационно-образовательного пространства для организации обучения студентов посвящено достаточно много работ [3–5]. С позиции организации сетевого обучения в распределенной программно-технической среде под информационно-образовательным пространством будем понимать «форму сетевого взаимодействия высших учебных заведений, интегрирующих научно-образовательный контент, к которому организуется коллективный доступ в процессе решения различных задач образовательной деятельности с использованием разнообразных программных сервисов» [9].

Характерными чертами современных сетевых форм взаимодействия образовательных организаций в рамках интегрированного информационно-образовательного пространства становятся:

- коллективный характер образовательного процесса с распределением обязанностей между организациями, установлением порядка реализации образовательной программы, определением характера и

объема ресурсов, используемых каждой организацией;

- создание гибких организационных структур, обеспечивающих динамическую реализацию образовательных программ для формирования конкретных профессиональных компетенций по индивидуальным и групповым траекториям обучения, особенно в программах дополнительного профессионального обучения;

- адаптация стратегии и тактики управления образовательными процессами, исходя из анализа тенденций развития рынка образовательных услуг и технологий;

- осуществление непрерывных инноваций в сфере развития образовательных технологий на основе применения передовых цифровых технологий.

В этих условиях возрастает актуальность применения цифровых технологий для автоматизации процессов динамического проектирования и реализации совместных образовательных проектов учебных заведений в сетевой среде. К таким цифровым технологиям в первую очередь относятся технологии искусственного интеллекта и машинного обучения, интеллектуальных многоагентных систем и семантического онтологического инжиниринга, на основе которых осуществляется инжиниринг современных сетевых образовательных структур.

1. Постановка задачи создания системы сетевого взаимодействия учебных заведений в информационно-образовательном пространстве

Образовательная программно-техническая платформа сетевого взаимодействия учебных заведений предоставляет коллективный доступ учебных заведений к совместно-используемому научно-образовательному контенту в процессе решения различных задач об-

разовательной деятельности с использованием различных программных сервисов. Научно-образовательный контент накапливается и организуется в распределенном хранилище учебных объектов. Программные сервисы можно рассматривать как функционал программно-технической платформы, который обеспечивает прежде всего:

1. динамическое построение программы обучения по запросу заказчика обучения на основе привлечения внешних образовательных ресурсов, наилучшим образом соответствующих его требованиям;

2. подготовку и согласование графика и режима обучения (очное, он-лайн, офф-лайн) между участниками образовательного процесса, представляющими разные учебные заведения.

Сетевое взаимодействие различных участников образовательного процесса представлено в виде диаграммы вариантов использования на рис. 1.

Реализация данной функциональности характеризуется динамичностью изменения ситуаций, соответствующих информационным потребностям и возможностям, необходимостью принимать оперативные решения, согласующиеся с целями и задачами каждого участника образовательного процесса. Сетевое взаимодействие участников образовательного процесса предполагает определение ролей:

1. заказчика обучения (клиента): заказчик формулирует требования к процессу обучения и его результатам;

2. организатора обучения: организатор строит программу обучения на основе требований клиента и подбирает необходимые для выполнения программы образовательные ресурсы, основная задача организатора при этом — согласование требований клиента и возможностей владельцев образовательных ресурсов;

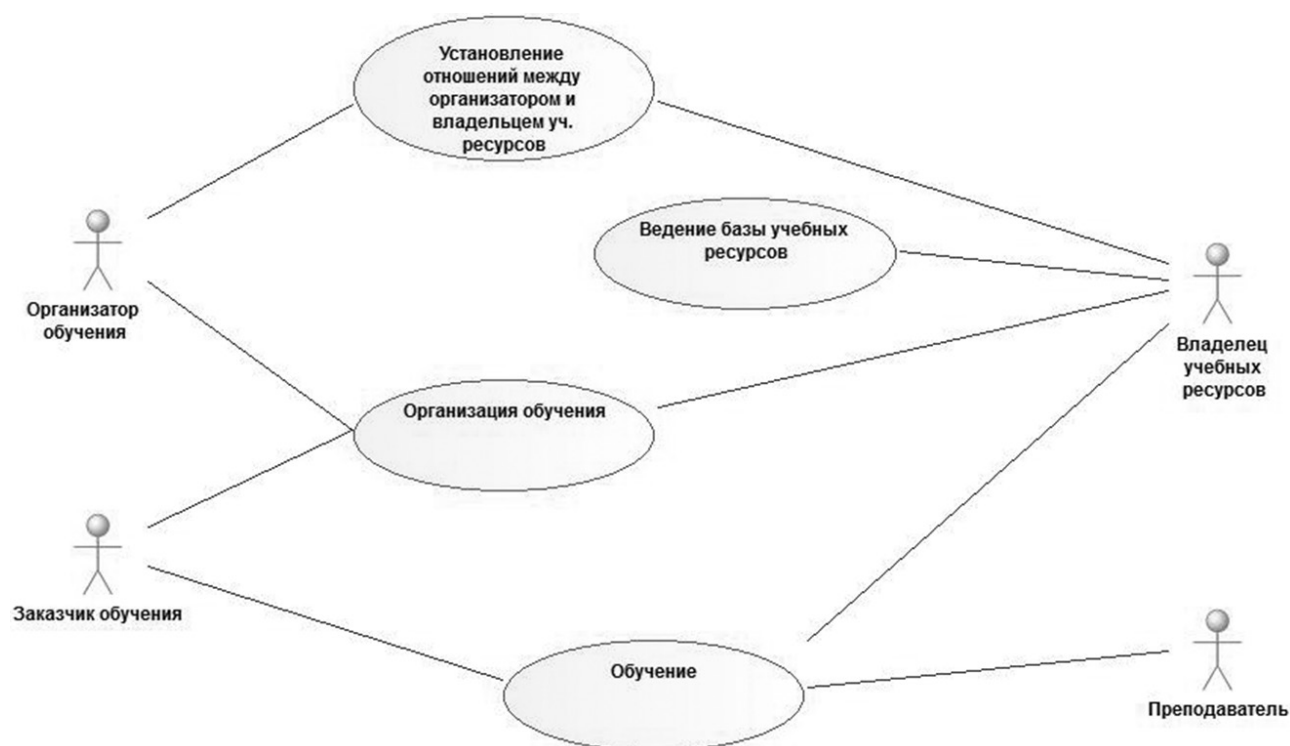


Рис. 1. Диаграмма вариантов использования процессов сетевого взаимодействия участников совместной образовательной деятельности

3. владельцев образовательных ресурсов: владельцы ресурсов разрабатывают, поддерживают и предоставляют в учебный процесс различные образовательные ресурсы — учебные занятия, проводимые преподавателями, учебно-методические ресурсы, тесты, линии он-лайн и офф-лайн консультаций и т.п.;

4. преподавателей: преподаватели осуществляют взаимодействие с клиентом (обучающимся) в рамках учебных мероприятий, организуемых владельцем образовательных ресурсов.

Для современных программно-технических образовательных платформ, реализующих концепцию информационно-образовательного пространства, характерным становится применение микросервисных архитектур, обеспечивающих разделение функциональности систем управления обучением LMS и систем управления контентом LCMS [6–9], что позволяет динамически обмениваться контентом, формировать про-

граммы и графики обучения. В связи с созданием таких интегрированных программно-технических платформ возникает целый ряд новых проблем, требующих своего решения, к которым относятся:

- Необходимо обеспечить полную семантическую интероперабельность. Современные платформы электронного обучения могут пересылать пользовательскую информацию, но не умеют «понимать» семантику информации, необходимую для анализа или для использования в различных средах, что вызывает необходимость онтологического представления семантики используемых знаний.

- Помимо обмена контентом образовательные платформы должны поддерживать обмен управленческими решениями между структурами, взаимодействующими в информационно-образовательном пространстве. В этом случае управление совместными образовательными процессами становится децентрализованным и динамическим в зави-

симости от оперативного изменения ситуации.

- Требуется обеспечение гибкого расширения функционала программно-технической платформы за счет использования инструментальных средств и стандартов поддержки подключаемости нового сервисов.

Некоторые работы по стандартизации [10] упоминают о необходимости создания комплекса сервисов, позволяющего динамически обнаруживать и компоновать соответствующие сервисы для того, чтобы разрешить пользовательский запрос. Для реализации перечисленных требований целесообразно использовать современные сервисные и/или многоагентные технологии [11–17]. Вместе с тем, следует рассмотреть основные различия между подходами по применению агентов и сервисов.

Сервис можно рассматривать как «примитивный» тип агента, имеющий триггерный тип управления, в отличие от интеллектуального агента, обладающего более ярко выраженными «антропоморф-

ными» свойствами. Интеллектуальный агент имеет более высокий уровень технологии управления, заключающийся во взаимодействии с окружающей средой в соответствии с определенной системой требований, в отличие от сервиса, взаимодействующего с окружающей средой по детерминированному алгоритму. В сервисно-ориентированной архитектуре сервис определяется как атомарная часть функционала системы, слабо связанная с другими сервисами, за счет их взаимодействия через стандартизированные интерфейсы и стандартизированные протоколы. При помощи сервисов обеспечивается разделение труда при выполнении определенной задачи в системе. Сервисы могут взаимодействовать друг с другом только по определенному алгоритму, предусматривающему дискретное количество состояний среды.

Важнейшей особенностью многоагентной системы является коллективное взаимодействие агентов. В многоагентных системах агент, не способный решить заданную задачу самостоятельно, может обратиться к другим агентам, либо для решения определенной задачи организуется кооперация многих агентов. При этом, агенты строят планы своих действий, основываясь как на своих возможностях, так и на планах и намерениях других агентов. В этой связи решается задача распределения ролей между участниками сетевых взаимодействий агентов, в процессе которой возникает необходимость:

- формирования совместных планов действий;
- учета интересов партнеров агента;
- синхронизации совместных действий;
- разрешения конфликта целей и интересов;
- конкурентного распределения совместных ресурсов;

- организации переговоров о совместных действиях;
- распознавание необходимости кооперации;
- выбора подходящего партнера;
- обучения поведению в коллективе;
- декомпозиции задач и разделения обязанностей;
- установления правил поведения в коллективе;
- исполнения совместных обязательств и др.

В агентской парадигме, агенты выполняют по отдельности простые функции, но при этом они обеспечивают решение сложной задачи в процессе кооперации. Один и тот же агент может участвовать в решении разных задач, и конфигурация агентского взаимодействия определяется динамически в зависимости от состояния окружающей среды, состояния агентов и уже выполненной работы.

Таким образом, можно сделать вывод, что парадигма многоагентных систем является наиболее подходящей для построения архитектуры образовательных платформ, реализующих информационно-образовательное пространство вследствие интеллектуальности и более сложного характера взаимодействия программных компонентов системы.

В отличие от ранее разработанного авторами программного прототипа поддержки информационно-образовательного пространства сетевого взаимодействия учебных заведений [2] новый прототип обеспечивает максимальную стандартизацию сетевого взаимодействия, упрощение и ускорение взаимодействия интеллектуальных программных за счет построения и применения единой программно-технической платформы, а не набора разнородных программных компонентов, как было в предыдущей версии системы. Каждый из участников процесса в рамках ИОП может рассматри-

ваться, как агент. В рамках данной работы будет рассмотрено создание и использование двух программных агентов – Организатора обучения и Владельца учебных ресурсов. При этом Заказчик будет рассматриваться как пользователь системы (взаимодействует с агентом Организатором обучения), а взаимодействие заказчика (обучающегося) и преподавателя в рамках процесса обучения выходят за границы постановки задачи. С концептуальной точки зрения создаваемая программно-техническая платформа обладает в большей степени свойствами универсальности применения для различных форм построения сетевых предприятий и организаций.

2. Архитектура системы сетевого взаимодействия учебных заведений в информационно-образовательном пространстве в программной среде JADE

Для построения многоагентной системы (МАС) в статье предлагается использовать широко известный набор стандартов Foundation for Intelligent Physical Agents (FIPA), разработанный комитетом IEEE Computer Society. Стандарты FIPA [18] регламентируют различные аспекты построения МАС – архитектуру системы и механизмы управления агентами системы, механизмы взаимодействия агентов системы посредством асинхронного обмена языковыми сообщениями (коммуникативные акты) и их обработки на основе предоставления сервисов. FIPA представляет в режиме открытого доступа набор спецификаций, определяющих различные компоненты многоагентной системы. Разработчик вправе использовать все, либо часть стандартов для реализации МАС.

Использование FIPA-стандартов имеет следующие преимущества:

1. Разные платформы, поддерживающие FIPA-стандарты, могут участвовать в одной MAC.

2. Разработка MAC на FIPA-совместимой платформе может происходить без необходимости детализации программной реализации, используя набор необходимых шаблонов.

Для разработки прототипа образовательной платформы как части ИОП с использованием многоагентных технологий проведен сравнительный анализ доступных инструментальных средств для разработки MAC.

В качестве потенциальных кандидатов для выбора программной среды реализации рассматривалось 7 фреймворков и библиотек: JADE (Java Agent Development Network) [19, 20], JACK Intelligent Agents [21], MadKIT [22], Cougaar [23], NetLogo [24], MASON [25], Repast Symphony [26], которые реализуют спецификации FIPA. Перечисленные инструментальные средства используют язык программирования Java, как правило, функционируют в кросс-платформенной среде и являются свободно распространяемым программным обеспечением. Среди инструментов построе-

ния MAC на основе FIPA наиболее широкие возможности предоставляет платформа и пакет разработки (framework) многоагентных систем JADE, разработанные исследовательским подразделением итальянской телекоммуникационной компании Telecom Italia (TI Lab). Несмотря на то, что система разработана телекоммуникационной компанией, она может применяться для решения широкого спектра задач — не только в области телекоммуникаций. JADE имеет хорошо проработанную документацию и типовые примеры использования, доступные на сайте проекта [20]. JADE включает в себя платформу развертывания и функционирования агентов MAC с распределенной гетерогенной архитектурой и библиотеку классов JAVA для разработки собственных агентов. Кроме того, в состав JADE входит набор инструментов для мониторинга, управления и тестирования работы агентов системы. В силу этих факторов JADE была выбрана в качестве инструментальной среды разработки многоагентной системы, поддерживающей информационно-образовательное

пространство сетевого взаимодействия учебных заведений (MAC ИОП).

Для реализации программного прототипа MAC ИОП разработке подлежит ряд механизмов функционирования системы на основе многоагентного подхода:

- интеллектуальных механизмов реализации принятия решений отдельных агентов;
- распределенной сетевой архитектуры системы, где каждый агент существует в обособленной программно-аппаратной среде (узлах сети) и при этом независим от этой среды;
- смеханизмов интеллектуального взаимодействия агентов системы на основе общей предметно-ориентированной онтологии.

Концепция архитектуры MAC ИОП представлена на рис. 2.

Агенты в JADE функционируют в среде базового JAVA-приложения, которое называется контейнером. Т.е. на каждом узле MAC агент может быть активизирован только в среде контейнера JADE. Несколько совместно функционирующих распределенных контейнеров с агентами

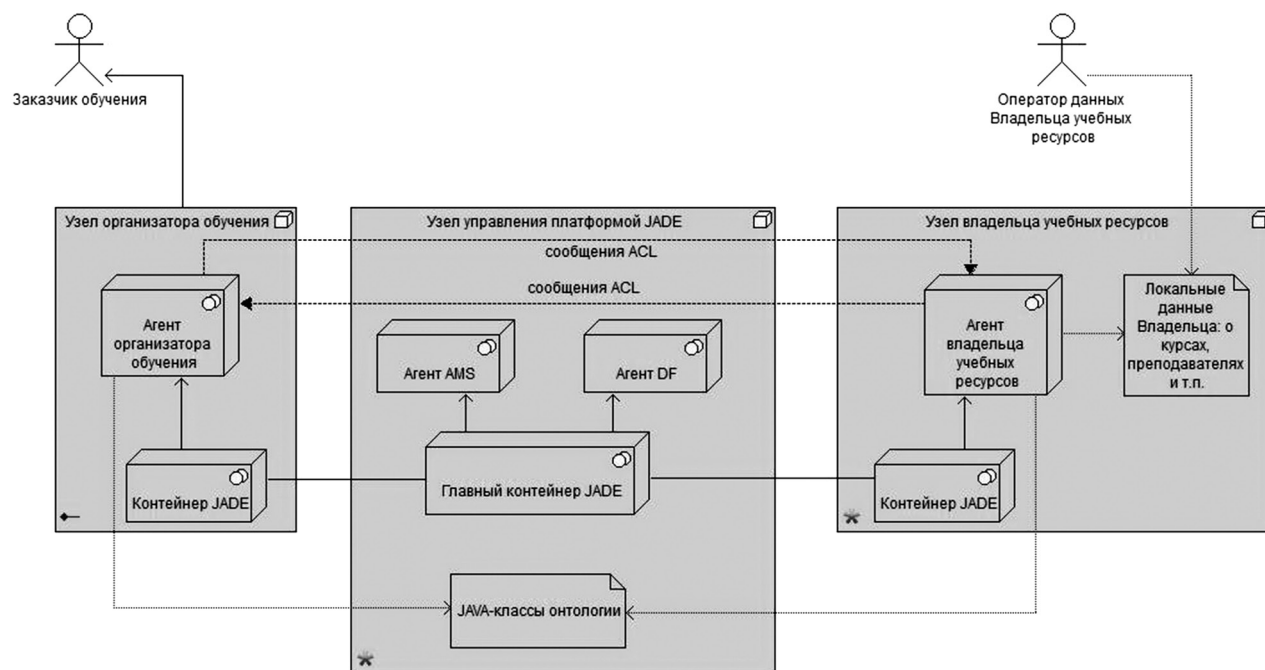


Рис. 2. Концепция архитектуры MAC ИОП

образуют платформу. Первый из запущенных контейнеров платформы считается главным и управляет работой всей системы. Он содержит два основополагающих механизма, обеспечивающих функционирование платформы и функционирующих как специальные агенты системы, – система управления агентами (AMS, Agent Management System) и служба каталога (DF, Directory Facilitator). AMS обеспечивает регистрацию и уникальное именование агентов платформы, а DF предоставляет сервис Желтых страниц, согласно FIPA, т.е. позволяет агентам объявлять свои сервисы для возможности поиска и запроса их другими агентами. Таким образом, MAC ИОП будет реализована на единой платформе JADE, где на сервере управления MAC будет активен главный контейнер с AMS и DF, а на узлах организаторов обучения и владельцев учебных ресурсов будут активны обычные контейнеры с соответствующими агентами системы.

Обмен сообщениями между агентами системы обеспечивают контейнеры системы, предоставляющие транспортный сервис (MTS, Message Transport Service). Сообщения, которыми обмениваются агенты имеют формат в соответствии с языком коммуникации между агентами JADE (ACL, Agent Communication Language) и содержат в себе транспортную информацию и полезное содержимое (контент) на одном из поддерживаемых язы-

ков представления контента (например, SL или LEAP).

Для того, чтобы содержимое сообщений могло одинаково восприниматься всеми агентами системы, оно должно содержать только термы, определенные в одной или нескольких онтологиях системы [20]. JADE поддерживает два варианта использования онтологий для представления семантики сообщений, передаваемых между агентами. Первый вариант предполагает наличие в архитектуре MAC отдельного сервера онтологий, который может хранить содержимое онтологий и выдавать его по запросу. Такой сервер не входит в состав платформы JADE и реализуется отдельно с помощью отдельного инструментария. Весь доступ к такому серверу онтологий осуществляется через специального агента на платформе JADE. Во втором случае все термы, составляющие онтологию, объявляются в виде JAVA-классов и используются затем всеми агентами для подготовки и разбора содержимого сообщений. В этом случае никакого хранилища онтологий в архитектуре MAC не нужно – онтология является частью самих агентов. В рассматриваемой здесь MAC будет использован именно этот вариант. Поэтому в архитектуре отсутствует хранилище онтологий, как отдельный элемент (рис.2).

Сценарий обмена сообщениями между агентами регламентирован протоколами взаимодействия между аген-

тами (IP, Interaction Protocol), которые построены на основе последовательности сообщений, содержащих стандартные коммуникативные акты. Процесс организации обучения может быть построен на основе Contract-Net IP, который подразумевает отсылку инициатором взаимодействия (в данном случае Организатором обучения) запроса на предложения (стандартный коммуникативный акт call-for-proposals, cfp) всем заинтересованным агентам (Владельцам учебных ресурсов, реализующих учебные курсы по требуемым компетенциям). После этого протокол подразумевает ожидание инициатором в течение определенного времени предложений от других участников взаимодействия; они могут выслать предложение (стандартный коммуникативный акт proposal), либо отказаться от сотрудничества (стандартный коммуникативный акт refuse). Рассмотрев полученные предложения, инициатор по каждому предложению может либо принять его (стандартный коммуникативный акт accept-proposal), либо отклонить (стандартный коммуникативный акт reject-proposal). Агенты, получившие одобрение своего предложения выполняют соответствующий сервис и по окончании работы, сообщают о результате (стандартный коммуникативный акт inform), либо о невозможности выполнения по какой-либо причине (стандартный коммуникативный акт failure).

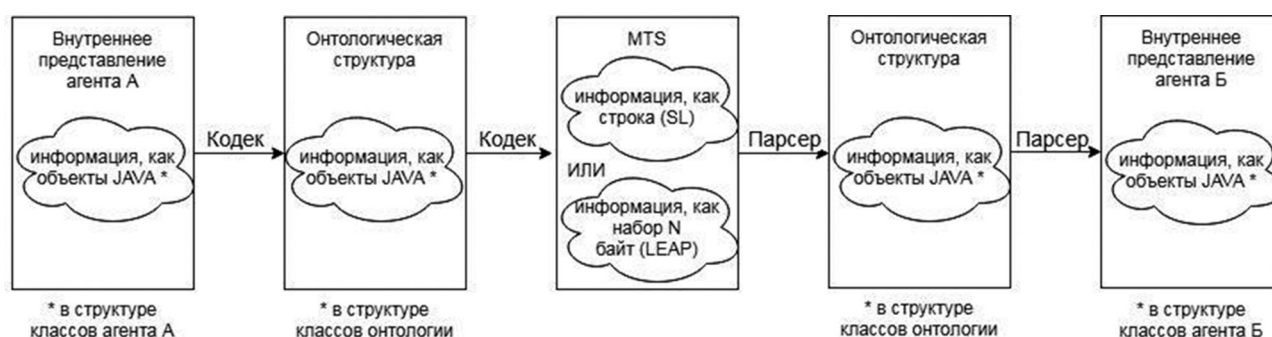


Рис. 3. Преобразование информации в процессе обмена сообщениями с использованием онтологий

Одним из центральных компонентов архитектуры МАС ИОП является онтология, которая обеспечивает понимание агентами семантики содержащихся в сообщениях концептов (понятий). При взаимодействии программных агентов друг с другом осуществляется обмен некоторой информацией посредством формируемых ACL сообщений. Внутри агента информация структурирована в виде JAVA-объектов, а внутри сообщения контент представлен в виде некоторого набора выражений на соответствующем языке (SL или LEAP). Таким образом, в процессе обмена сообщениями информация видоизменяется несколько раз – кодируется и декодируется (рис. 3). Кроме того, осуществляется валидация сообщений при получении на предмет соответствия онтологии, которой пользуется получатель сообщения. В данном случае онтология представляет собой общий словарь, необходимый для формирования и «понимания» сообщений.

3. Реализация онтологии МАС ИОП

Для того чтобы сформировать онтологию, полностью отвечающую требованиям проектируемой МАС, необходимо

проанализировать существующие варианты взаимодействия между агентами, оценить структуру и типы сообщений, а также виды активностей агентов и описать схему следующих элементов онтологии:

1. Концепты-сущности (j.O:Concept). Наполняют онтологию предметной области и обычно используются для формирования предикатов и запросов. Концепты-сущности включают описание некоей сущности, ее атрибутов и статические отношения с другими объектами). Концепты-сущности могут объединяться в агрегаты (последовательности). К концептам-сущностям в МАС ИОП относятся: Компетенции (competence) и элементы компетенций (Знания – knowledge, Умения – ability, Навыки – skills) и др.

2. Предикаты (j.O:Predicate). Представляют собой описание фактов о предметной области, описывают динамические отношения между объектами и могут быть истинными или ложными. Обычно предикаты используются в случае отправки сообщений типа INFORM («верно, что...» или QUERY-IF («верно ли, что... ?»). К предикатам онтологии МАС ИОП относятся: Содержание ЗУН в курсах (StudiedInCourse), Расписание

курсов (CourseTimetable), Содержание ЗУН в компетенциях (ContainsCompetenceElement).

3. Концепты-действия агентов (j.O:Agent-action). Входят в структуру онтологии задач и в отличие от концептов-сущностей могут использоваться самостоятельно в случае отправки сообщения типа REQUEST (запрос на выполнение действия). К концептам-действиям относят: Назначение курса в график обучения (PlanCourse)

4. Специальные концепты. Например, такие как AID.

Концептуальная структура онтологии МАС ИОП в графическом представлении редактора онтологий Protégé [27] и описание ограничений и условий для классов на примере предиката CourseTimetable показаны на рис. 4.

На основе указанных структурных элементов могут быть сформированы идентифицирующие референциальные выражения. Эти выражения используются для поиска сведений, сущностей, для которых заданный предикат является истинным, в сообщениях типа QUERY_REF (отбор по условию).

Структура референциального выражения предполагает объявление переменных на месте одного или нескольких аргументов предиката.

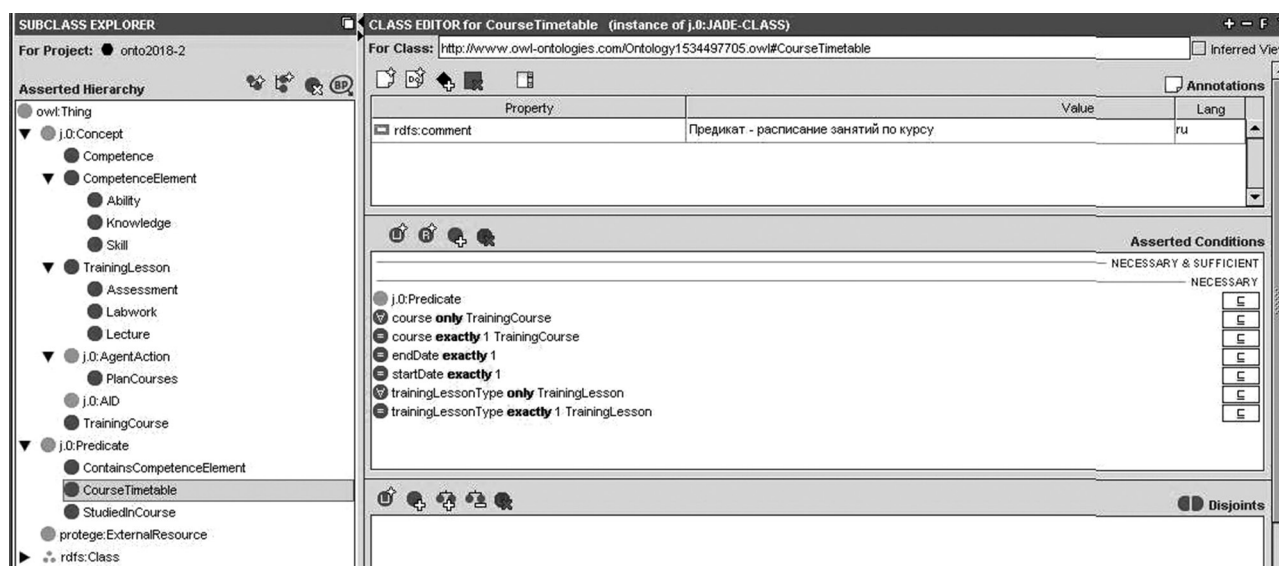


Рис. 4. Концептуальная структура онтологии МАС ИОП в Protégé

```
(all ?x
  (ContainsCompetenceElement
    (Competence :competenceName "Организационное и методологическое
обеспечение проверки реализации запросов на изменение")
    ?x)
```

Рис. 5. Запрос на отбор ЗУН для заданной компетенции на языке SL

```
AbsConcept absCompetence = ontology.fromObject(competence);
AbsVariable x = new AbsVariable("x", EduOntology.COMPETENCEELEMENT);
AbsPredicate absCCE = new AbsPredicate(EduOntology.CONTAINSCOMPETENCEELEMENT);
absCCE.set(EduOntology.COMPETENCE, absCompetence);
absCCE.set(EduOntology.COMPETENCEELEMENT, x);
AbsIRE absIota = new AbsIRE(SLVocabulary.IOTA);
absIota.setVariable(x);
absIota.setProposition(absCCE);
getContentManager().fillContent(msg, absIota);
send(msg);
```

Рис. 6. Формирование запроса на отбор ЗУН для заданной компетенции в агенте

Для отбора сведений о ЗУН, соответствующих некоторой компетенции, необходимо определить в онтологии предикат, обозначающий связь компетенции и ЗУН, в качестве одного аргумента этого предиката включить в запрос описание компетенции, а в качестве второго аргумента указать переменную. Результатом выполнения запроса должен стать набор ЗУН, при условии подстановки которых в представленный шаблон, предикат будет принимать значение «Истина».

Запрос на языке SL имеет следующий вид, представленный на рис. 5.

Однако в рамках JADE формирование и кодирование референциального выражения заключается в описании ряда специализированных объектов, передаваемых в качестве параметров запроса, переменной и посылки, и только после этого используется преобразователь (кодек) для формирования строки запроса на языке SL. Фрагмент формирования запроса, рассмотренного выше, в виде набора JAVA-объектов представлен на рис. 6.

Порядок создания онтологии в JADE также подразумевает описание на языке JAVA соответствующих классов (концепты, предикаты, действия агентов) и методов чтения/записи, унаследованных от стандартных классов фрейм-

ворка, а затем включение их в класс, определяющий онтологию BeanOntology [20].

На рис. 7 приведен пример описания класса Competence.

Описанные таким образом классы могут быть объединены

в пакет (например, «examples.content.competence.elements») или включены в онтологию по отдельности (например, для того, чтобы организовать иерархию понятий). Оба варианта представлены на рис. 8.

Большое количество взаимосвязанных классов, а также атрибутов, которые необходимо формально описать, увеличивает трудоемкость ручной подготовки онтологии MAC, а также увеличивает риск ошибки, внесенной при описании онтологии вручную. Для автоматизации создания онтологии используется плагин Ontology Bean Generator (автор Крис ван Аарт) [28], который расширяет возможности редактора онтологий Protégé. Плагин позволяет генерировать соответствующие фреймворку JADE

```
// Объявление концепта
public class Competence implements Concept {

// Объявление атрибутов
private String CompetenceName;
private Integer CompetenceLevel;

// Объявление методов
public String getCompetenceName () {
    return CompetenceName;
}
public void setCompetenceName (String CompetenceName) {
    this.CompetenceName = CompetenceName;
}
public Integer getCompetenceLevel () {
    return CompetenceLevel;
}
public void setCompetenceLevel (Integer CompetenceLevel) {
    this.CompetenceLevel = CompetenceLevel;
}
}
```

Рис. 7. Описание класса Competence

```
// Объявление онтологии
public class CompetenceOntology extends BeanOntology {
    private static Ontology theInstance = new CompetenceOntology(CompetenceOntology2018);
    public static Ontology getInstance() {
        return theInstance;
    }
    private CompetenceOntology(String name) {
        super(name);
        try {
            // Добавление класса Competence
            add(Competence.class);
            // Добавление классов из пакета
            add("examples.content.competence.elements ");
        }
        catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

Рис. 8. Описание онтологии

онтологии на основе описаний онтологий в формате RDF(S) и OWL, а также в собственном формате Protégé.

4. Реализация программного прототипа МАС ИОП

Рассмотрим реализацию программного прототипа МАС ИОП с точки зрения агентного взаимодействия. Схема взаимодействия агентов в рамках прецедента Организации обучения представлена на рис. 9 в виде диаграммы последовательностей.

Процесс организации обучения, показанный на рис. 9 реализован в JAVA-коде соответствующих агентов JADE:

1. агента *Организатора обучения*, выступающего в качестве инициатора взаимодействия — class LearningOrgAgent. Процесс организации обучения начинается с запуска на выполнение этого агента. Данный агент после запуска должен собрать необходимые

данные о параметрах обучения у *Заказчика*; найти *агентов-владельцев учебных ресурсов*, предоставляющих учебные ресурсы по заданным компетенциям, и подобрать среди них наиболее подходящих под требования заказчика; сделать заказ на обучение у выбранных агентов, получить от них расписание обучения и выслать расписание заказчику. Агент реализует следующие методы:

- метод BestLearner() — определяет *агента-владельца учебных ресурсов* с наиболее подходящим под требования заказчика предложением на обучение;
- метод SendShedule() — высылает расписание заказчику на адрес его электронной почты.

2. агента *Владельца учебных ресурсов*, выступающего в качестве участника этого взаимодействия — class EducatResOwner. Агенты этого типа могут регистрироваться в системе каждым владельцем учебных ресурсов

(например, ВУЗом) в любое время для возможности предоставления собственных услуг *Заказчику* (обучение по предоставляемым учебным курсам). В задачи этого агента входит регистрация в DF в качестве сервисов компетенций, по которым агент обладает учебными ресурсами (за рамками рассматриваемого процесса); подбор и предложение подходящих учебных курсов по запросу других агентов (организатора обучения); формирование расписания обучения по выбранным курсам. Агент реализует следующие методы:

- метод SearchEdRes(ZUN) — возвращает учебные ресурсы, реализующие указанный ЗУН;
- метод SheduleLearning(Course) — планирует обучение по указанному курсу в сетке обучения *владельца учебных ресурсов* в соответствии с существующей загрузкой ресурсов (преподавателей, учебных аудиторий) и формирует расписание по курсу.

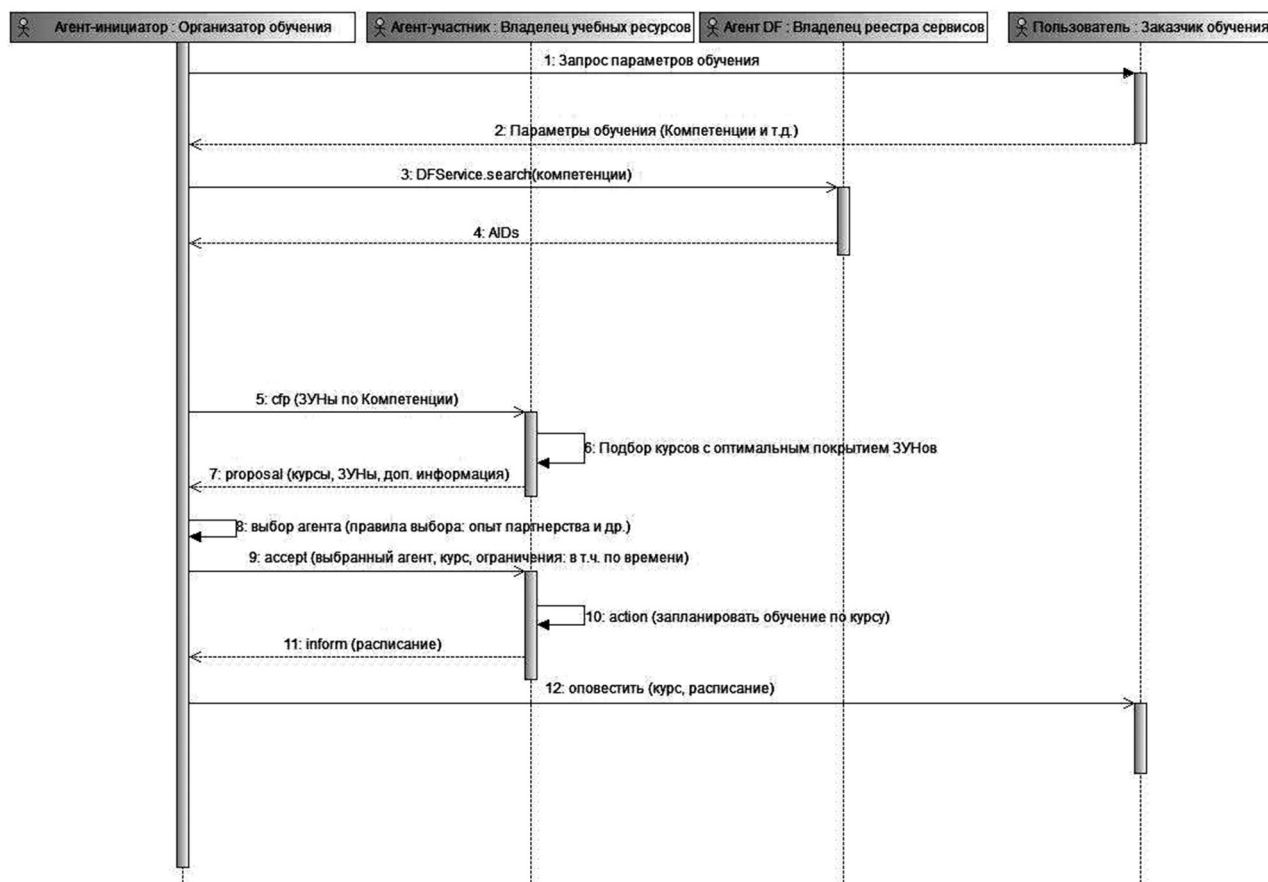


Рис. 9. Диаграмма последовательностей взаимодействия агентов в процессе организации обучения

На рис.9 отражены следующие этапы взаимодействия этих двух агентов в рамках процесса организации обучения для *Заказчика*:

1. На начальном этапе (шаги 1–2), при запуске *агента-организатора обучения* на выполнение, он опрашивает *Заказчика* на предмет требований к параметрам обучения. Прежде всего выявляются необходимые компетенции, которыми должен обладать обучающийся в результате обучения. Кроме того, могут быть заданы дополнительные требования к форме обучения (очная, электронная и т.п.), режиму обучения (длительность, интенсивность, время занятий и т.п.), преподавателям (наличие ученой степени, практический опыт и т.п.), стоимости обучения. Это поведение агента реализовано в его методе `setup()`.

2. На следующем этапе (шаги 3–4) *агент-организатор* подбирает в сервисе «желтых страниц», предоставляемом системным агентом DF, агентов-владельцев учебных ре-

сурсов, специализирующихся по направлениям соответствующих компетенций. Несмотря на то, что DF – такой же агент, как и другие, и взаимодействие с ним ведется с помощью ACL-сообщений, в JADE вся процедура запроса и получения ответа от DF реализована в одном методе агента `DFService.search()`. Метод возвращает идентификаторы агентов в системе (AID).

Подбор агентов реализован в методе `setup()` как это представлено на рис. 10.

3. Далее (шаг 5) *агент-организатор* инициирует взаимодействие с найденными в DF *агентами-владельцами ресурсов* в соответствии со стандартным протоколом FIPA Contract-Net и направляет этим агентам запрос на предложение (коммуникативный акт `call-for-proposal`) учебных курсов. В контенте запроса в соответствии с онтологией MAC указываются предикаты, устанавливающие соответствие ЗУНов (знания, умения и навыки) выбранной заказчиком компетенции.

4. На шаге 6 *агенты-владельцы учебных ресурсов*, получившие запрос, подбирают учебные курсы, наиболее полно охватывающие необходимые ЗУНы с помощью метода `SearchEdRes(ZUN)`, и на шаге 7 высылают ответное сообщение со своими предложениями по обучению (коммуникативный акт *proposal*), включающими ссылку на курс, охватываемые им ЗУНы и доп. информацию о форме обучения по курсу, авторах и преподавателях курса и т.п.

5. После окончания отведенного времени из числа полученных предложений, *агент-организатор обучения* с помощью метода `BestLearner()` выбирает (шаг 8) в соответствии с собственной базой знаний наилучшие предложения – по степени соответствия требованиям *Заказчика*, опыту предыдущего сотрудничества и т.п., после чего высылает (шаг 9) соответствующим *агентам-владельцам учебных ресурсов* сообщение с согласием на их предложение (коммуникативный акт *accept-proposal*). Согласие означает необходимость только запланировать обучение по курсу, т.к. сам процесс обучения лежит за границами системы.

6. *Агент-владелец учебных ресурсов*, получивший согласие, с помощью метода `SheduleLearning(Course)` формирует (шаг 10) расписание учебных мероприятий по курсу, исходя из предоставленных требований и существующего графика учебного процесса и отправляет (шаг 11) расписание в качестве результата (коммуникативный акт *inform*) организатору. На этом выполнение стандартного протокола Contract-Net окончено.

7. После получения расписания *агент-организатор* доводит (шаг 12) это расписание до сведения *Заказчика* с помощью метода `SendShedule()` (высылает на адрес электронной почты заказчика).

```
public class LearningOrgAgent extends Agent {
    private Vector EdResOwnerAgents;
    ...
    protected void setup() {
        ...
        Scanner in = new Scanner(System.in);
        System.out.print("Input Competition for your learning:");
    };
    String competition = in.nextLine();
    ...
    DFAgentDescription template = new DFAgentDescription();
    ServiceDescription sd = new ServiceDescription();
    sd.setType(competition);
    template.addServices(sd);
    try {
        DFAgentDescription[] result = DFService.search(myAgent,
        template);
        EdResOwnerAgents.clear();
        for (int i = 0; i < result.length; ++i) {
            EdResOwnerAgents.addElement(result[i].getName());
        }
    }
    ...
}
```

Рис. 10. Описание подбора агентов

На этом процесс завершается и обучение в соответствии с расписанием проводится уже вне рамок МАС.

Заключение

Реализация программного прототипа многоагентной системы сетевого взаимодействия учебных заведений продемонстрировала возможность гибкого формирования совместных образовательных программ и

расписаний в соответствии с динамическими потребностями в освоении переменных наборов профессиональных компетенций. При этом автоматизация подобных взаимодействий с использованием интеллектуальных программных агентов позволяет повысить оперативность и качество решения подобных задач.

Разработанный программный прототип многоагентной системы может использоваться

не только в целях реализации программ сетевого обучения, но с некоторыми изменениями для любых экономических объектов, специализирующихся на сложном построении бизнес-процессов обслуживания потребителей, например, в сфере мультимодальных транспортных перевозок, в туристическом бизнесе, разработке инновационных проектов в различных сферах экономической деятельности.

Литература

1. Федеральный закон «Об образовании в Российской Федерации» от 29.12.2012 № 273-ФЗ.
2. Тельнов Ю.Ф., Казаков В.А., Данилов А.В. Программная реализация информационно-образовательного пространства на основе многоагентной технологии и онтологического подхода // Открытое образование. 2015. № 6. С. 73–82.
3. Виноградова Е. А. Создание единого информационно-образовательного пространства в образовательном учреждении. ИКТ – новая технология управления в образовательном учреждении – Педагогическая мастерская [Электронный ресурс] Режим доступа: <http://открытыйурок.рф/статьи/538755/> (Дата обращения: 28.01.2018)
4. Ширина Н.В. Информационно-образовательное пространство школы – Педагогическая мастерская [Электронный ресурс] Режим доступа: <http://открытыйурок.рф/статьи/503319/>
5. Зиндер Е.З. Новая парадигма инжиниринга предприятий и требования к новым ИТ-специальностям // Современные информационные технологии и ИТ-образование. 2013. № 9. [Электронный ресурс] Режим доступа: <https://cyberleninka.ru/article/n/novaya-paradigma-inzhiniringa-predpriyatiy-i-trebovaniya-k-novym-it-spetsialnostyam>
6. Андреев А.А. Российские открытые образовательные ресурсы и массовые открытые дистанционные курсы // Высшее образование в России. 2014. № 6. С. 150–155.
7. Днепровская Н.В. Система управления знаниями как основа смарт-обучения // Открытое образование. 2018. № 22 (4). С. 42–52. DOI: <https://doi.org/10.21686/1818-4243-2018-4-42-52>
8. Hai Zhang, Takanori Maesako A Framework of Learner Development Ecosystem for Designing a Ubiquitous Educational Informational Infrastructure // Journal of Software. 2009. Vol. 4. № 2.
9. Тельнов Ю.Ф., Гаспарян М.С., Диго С.М., Казаков В.А., Смирнова Г.Н., Ярошенко Е.В., Трёмбач В.М. Реализация процессов учебно-методического обеспечения в интегрированном информационно-образовательном пространстве на основе сервисной архитектуры // Статистика и экономика. 2015. № 1. С. 198–206.
10. Feier C., Polleres A., Roman D., Domingue J., Stollberg M., Fensel D. Towards Intelligent Web Services: The Web Service Modeling Ontology (WSMO) // Proc. of the Int'l Conf on Intelligent Computing, ICIC. 2005. P. 789–798.
11. Трёмбач В.М. Многоагентная система для решения задач целенаправленного поведения. // В кн. Четырнадцатая национальная конференция по искусственному интеллекту с международным участием КИИ 2014 (24–27 сентября 2014 г., Казань). Труды конференции. Т.1. Казань: РИЦ «Школа», 2014. С. 344–353.
12. Трёмбач В.М. Когнитивный подход к созданию интеллектуальных модулей организационно-технических систем // Открытое образование. 2017. № 2. С. 78–87.
13. Даггер Д., О'Коннор А., Лавлесс С., Уолш Э., Уэйд В. Сервисные платформы электронного обучения: от монолитных систем к гибким сервисам // Открытые системы. СУБД. 2007. № 7. [Электронный ресурс] Режим доступа: <https://www.osp.ru/os/2007/07/4392612/>
14. IBM developerWorks Россия: SOA и Web-сервисы [Электронный ресурс] Режим доступа: <http://www.ibm.com/developerworks/ru/webservices>.
15. Wooldridge M. An introduction to MultiAgent Systems. 2nd edition. Hoboken: Wiley, 2009. 484 p.
16. Городецкий В.И., Грушинский М.С., Хабалов А.В. Многоагентные системы (обзор) // Новости искусственного интеллекта. М.: Изд-во ЦНИЭИуголь, 1998. № 2. 196 с.
17. Гаврилова Т.А., Кудрявцев Д.В., Муромцев Д.И. Инженерия знаний. Модели и методы. Учебник. СПб.: Издательство «Лань», 2016. 324 с.
18. Foundation for Intelligent Physical Agents – Specifications [Электронный ресурс] Режим

доступа: <http://www.fipa.org/specs/fipa00023/SC00023J.html>

19. Bellifemine F., Caire G., Greenwood D. Developing multi-agent systems with JADE. 1 edition. Hoboken: Wiley, 2007. 300 p.

20. JADE (Java Agent Development Network) [Электронный ресурс] Режим доступа: <http://jade.tilab.com>

21. Jack Intelligent Agents [Электронный ресурс] Режим доступа: <http://www.agent-software.com.au/products/jack/>

22. MadKIT [Электронный ресурс] Режим доступа: <http://www.madkit.net/madkit/>

23. Cougaar [Электронный ресурс] Режим доступа: <http://cougaar.org>

24. NetLogo [Электронный ресурс] Режим доступа: <https://ccl.northwestern.edu/netlogo/>

25. MASON [Электронный ресурс] Режим доступа: <https://cs.gmu.edu/~eclab/projects/mason/>

26. Repast Symphony [Электронный ресурс] Режим доступа: <https://repast.github.io>

27. Protégé [Электронный ресурс] Режим доступа: <https://protege.stanford.edu/>

28. OntologyBeanGenerator [Электронный ресурс] Режим доступа: <https://protegewiki.stanford.edu/wiki/OntologyBeanGenerator>.

References

1. Federal Law "On Education in the Russian Federation" dated December 29, 2012 N273-FZ. (In Russ.)

2. Tel'nov YU.F., Kazakov V.A., Danilov A.V. Software implementation of information and educational space based on multi-agent technology and ontological approach. *Otkrytoye obrazovaniye = Open Education*. 2015; 6: 73-82. (In Russ.)

3. Vinogradova E. A. Sozdaniye edinogo informatsionno-obrazovatel'nogo prostranstva v obrazovatel'nom uchrezhdenii. IKT - novaya tekhnologiya upravleniya v obrazovatel'nom uchrezhdenii - Pedagogicheskaya masterskaya = Creating a unified information and educational space in an educational institution. ICT - a new management technology in an educational institution - Pedagogical workshop [Internet] Available from: <http://otkrytyy-urok.rf/stat'i/538755/> (cited: 28.01.2018) (In Russ.)

4. SHirshina N.V. Informatsionno-obrazovatel'noye prostranstvo shkoly - Pedagogicheskaya masterskaya = Information and educational space of the school - Pedagogical workshop [Internet] Available from: <http://otkrytyy-urok.rf/stat'i/503319/> (In Russ.)

5. Zinder E.Z. New paradigm of enterprise engineering and requirements for new IT specialties. *Sovremennyye informatsionnyye tekhnologii i IT-obrazovaniye = Modern information technologies and IT education*. 2013; 9. [Internet] Available from: <https://cyberleninka.ru/article/n/novaya-paradigma-inzhiniringa-predpriyatiy-i-trebovaniya-k-novym-it-spetsialnostyam> (In Russ.)

6. Andreyev A.A. Russian open educational resources and mass open distance courses. *Vyssheye obrazovaniye v Rossii = Higher education in Russia*. 2014; 6: 150-155. (In Russ.)

7. Dneprovskaya N.V. Knowledge Management System as the Basis for Smart Learning. *Otkrytoye obrazovaniye = Open Education*. 2018; 22(4): 42-52. DOI: <https://doi.org/10.21686/1818-4243-2018-4-42-52> (In Russ.)

8. Hai Zhang, Takanori Maesako A Framework of Learner Development Ecosystem for Designing a Ubiquitous Educational Informational Infrastructure. *Journal of Software*. 2009; 4(2).

9. Tel'nov YU.F., Gasparian M.S., Digo S.M., Kazakov V.A., Smirnova G.N., YAroshenko E.V., Trembach V.M. Implementation of the teaching and methodical support processes in the integrated information and educational space based on the service architecture. *Statistika i ekonomika = Statistics and Economics*. 2015; 1: 198-206. (In Russ.)

10. Feier C., Polleres A., Roman D., Domingue J., Stollberg M., Fensel D. Towards Intelligent Web Services: The Web Service Modeling Ontology (WSMO). *Proc. of the Int'l Conf on Intelligent Computing, ICIC*. 2005: 789-798.

11. Trembach V.M. Mnogoagentnaya sistema dlya resheniya zadach tselenapravlennoye povedeniya = Multi-agent system for solving tasks of purposeful behavior. In: *CHetyrnadtsataya natsional'naya konferentsiya po iskusstvennomu intellektu s mezhdunarodnym uchastiyem KII 2014 Fourteenth National Conference on Artificial Intelligence with International Participation KII 2014* (September 24-27, 2014, Kazan). Conference proceedings. Vol.1. Kazan: RIC «School»; 2014: 344-353. (In Russ.)

12. Trembach V.M. Cognitive approach to the creation of intelligent modules of organizational and technical systems. *Otkrytoye obrazovaniye = Open Education*. 2017; 2: 78-87. (In Russ.)

13. Dagger D., O'Konner A., Lavless S., Uolsh E., Ueyd V. E-learning service platforms: from monolithic systems to flexible services. *Otkrytyye sistemy = Open Systems. DBMS*. 2007; 7. [Internet] Available from: <https://www.osp.ru/os/2007/07/4392612/> (In Russ.)

14. IBM developerWorks Rossiya: SOA i Web-servisy = IBM developerWorks Russia: SOA and Web services [Internet] Available from: <http://www.ibm.com/developerworks/ru/webservices>. (In Russ.)

15. Wooldridge M. An introduction to Multi-Agent System: 2nd edition. Hoboken: Wiley; 2009. 484 p.

16. Gorodetskiy V.I., Grushinskiy M.S., KHabalov A.V. Mnogoagentnyye sistemy (obzor). *Novosti iskusstvennogo intellekta = Multi-agent systems (review)*. News of artificial intelligence. Moscow: Publishing house CNIEIugol; 1998: 2. 196 p. (In Russ.)

17. Gavrilova T.A., Kudryavtsev D.V., Muromtsev D.I. Inzheneriya znaniy. Modeli i metody. Uchebnik. = Knowledge Engineering. Models and methods. Textbook. Saint Petersburg: Publishing house «Lan»; 2016. 324 p. (In Russ.)
18. Foundation for Intelligent Physical Agents – Specifications [Internet] Available from: <http://www.fipa.org/specs/fipa00023/SC00023J.html>
19. Bellifemine F., Caire G., Greenwood D. Developing multi-agent systems with JADE. 1 edition. Hoboken: Wiley; 2007. 300 p.
20. JADE (Java Agent Development Network) [Internet] Available from: <http://jade.tilab.com>
21. Jack Intelligent Agents [Internet] Available from: <http://www.agent-software.com.au/products/jack/>

22. MadKIT [Internet] Available from: <http://www.madkit.net/madkit/>
23. Cougaar [Internet] Available from: <http://cougaar.org>
24. NetLogo [Internet] Available from: <https://ccl.northwestern.edu/netlogo/>
25. MASON [Internet] Available from: <https://cs.gmu.edu/~eclab/projects/mason/>
26. Repast Symphony [Internet] Available from: <https://repast.github.io>
27. Protégé [Internet] Available from: <https://protege.stanford.edu/>
28. OntologyBeanGenerator [Internet] Available from: <https://protegewiki.stanford.edu/wiki/OntologyBeanGenerator>.

Сведения об авторах

Юрий Филиппович Тельнов

*Заведующий кафедрой Прикладной информатики и информационной безопасности
Российский экономический университет
им. Г.В. Плеханова,
Москва, Россия
Эл. почта: Telnov.YUF@rea.ru
Тел.: +7(495)800-12-00 (д. 2562)*

Андрей Владимирович Данилов

*Старший преподаватель кафедры Прикладной информатики и информационной безопасности
Российский экономический университет
им. Г.В. Плеханова, Москва, Россия
Эл. почта: Danilov.AV@rea.ru
Тел.: +7(495)800-12-00 (д. 2063)*

Рустэм Ильясович Дивеев

*Консультант
ООО Glowbyte Consulting, Москва, Россия
Эл. почта: Rustem.Diveev@gmail.ru
Тел.: +7(968)446-37-20*

Василий Александрович Казаков

*Ведущий научный сотрудник НИИ
«Стратегические информационные технологии»
Российский экономический университет
им. Г.В. Плеханова, Москва, Россия
Эл. почта: Kazakov.VA@rea.ru
Тел.: +7(495)800-12-00 (д. 2063)*

Елена Валерьевна Ярошенко

*Доцент кафедры Прикладной информатики и информационной безопасности
Российский экономический университет
им. Г.В. Плеханова, Москва, Россия
Эл. почта: YAroshenko.EV@rea.ru
Тел.: +7(495)800-12-00 (д. 2063)*

Information about the authors

Yuriy F. Telnov

*Dr. Sci. (Economics), Professor, Head of the
Department of applied information technologies and
information security
Plekhanov Russian University of Economics,
Moscow, Russia
E-mail: telnov.yuf@rea.ru
Tel.: +7(495)800-12-00 (ext. 2562)*

Andrey V. Danilov

*Senior Lecturer of the Department of Applied
Information Technologies and Information Security
Plekhanov Russian University of Economics,
Moscow, Russia
E-mail: danilov.av@rea.ru
Tel.: +7(495)800-12-00 (ext. 2063)*

Rustem I. Diveev

*Consultant
Glowbyte Consulting, Moscow, Russia
E-mail: Rustem.Diveev@gmail.ru
Tel.: +7(968)446-37-20*

Vasiliy A. Kazakov

*Ph.D. in Economics
Lead Researcher of the Research Institute «Strategic IT»
Plekhanov Russian University of Economics,
Moscow, Russia
E-mail: kazakov.va@rea.ru
Tel.: +7(495)800-12-00 (ext. 2063)*

Elena V. Yaroshenko

*Associate Professor of the Department of Applied
Information Technologies and Information Security
Plekhanov Russian University of Economics,
Moscow, Russia
E-mail: YAroshenko.EV@rea.ru
Tel.: +7(495)800-12-00 (ext. 2063)*