

МНОГОАГЕНТНАЯ МОДЕЛЬ ПРОЦЕССА ОБУЧЕНИЯ СТУДЕНТОВ НА КАФЕДРАЛЬНОМ УРОВНЕ

Федяев О. И., Жабская Т. Е., Грач Е.
Кафедра ПМиИ ДонНТУ
fedyaev@r5.dgtu.donetsk.ua

Abstract

This work investigates the use of agent-oriented technology to develop a computer environment for teaching students at a department. This model describes a department as an intelligent distributed system.

Введение

Учебный процесс в техническом ВУЗе обеспечивается его подразделениями с жесткой иерархической структурой (ректорат, факультеты, кафедры). На нижнем кафедральном уровне обучение профессионально ориентировано, распределено в пространстве, регламентировано по времени, динамично по содержанию и поэтому, как процесс подготовки инженерных кадров, является сложным для создания современных компьютерных средств распределённого обучения.

Установленные в настоящее время правила получения образования не позволяют своевременно реагировать на изменение требований рынка, в полной мере учитывать индивидуальные возможности и желания студентов в освоении дисциплины за ускоренное время (экстерном), предполагают обязательное личное присутствие преподавателя на всех этапах передачи и контроля усвоения знаний, жесткую привязку студентов к расписанию занятий. Студенту не всегда разрешают работать по индивидуальному графику, поскольку без общения с преподавателем, несмотря на полноценное учебно-методическое обеспечение, практически сложно получить хороший уровень знаний по предметам. Современная тенденция в инженерном образовании характеризуется внедрением индивидуальных образовательных схем, в полной мере отвечающих быстрым изменениям конъюнктуры рынка. Поэтому классические схемы централизованного управления образованием с жесткой структурой должны позволять трансформироваться в более гибкие схемы [1].

В статье рассматривается задача построения новой модели процесса обучения студентов дисциплинам, которая на основе сохранения близких к реально существующим взаимоотношениям между участниками учебного процесса и предоставления возможности автономного и распределенного выполнения учебно-методических обязанностей позволит повысить

децентрализованность и индивидуальность работы всех участников процесса обучения на кафедральном уровне.

1. Организация процесса обучения на основе агентно-ориентированного подхода

В учебном процессе, выполняемом кафедрой, участвуют следующие субъекты: профессорско-преподавательский состав кафедры (заведующий кафедрой M , лекторы L_1, L_2, \dots, L_n , ассистенты P_1, P_2, \dots, P_k), лаборанты (Y_1, Y_2, \dots, Y_m), студенты (X_1, X_2, \dots, X_n). На каждом отрезке времени (семестре) студенты изучают предусмотренные учебным планом специальности дисциплины (D_1, D_2, \dots, D_m). Изучение каждой дисциплины включает прослушивание лекций ($Lect_1, Lect_2, \dots, Lect_k$), выполнение лабораторных работ, возможно выполнение курсовой работы и сдачу экзамена (зачёта). Для успешной учебы студентам необходимо посещать лекции и общаться с преподавателями на практических и лабораторных занятиях, т. е. существуют установленные взаимодействия и взаимоотношения между субъектами учебного процесса.

Учебный процесс может быть описан следующими компонентами:

$$УП = (S, K, R, Act, I, T, U),$$

где $S = \{M, (X_1, X_2, \dots, X_n), ((L_1, L_2, \dots, L_n), (P_1, P_2, \dots, P_k)), (Y_1, Y_2, \dots, Y_m)\}$ – множество субъектов учебного процесса; K – среда (кафедра), в которой функционирует данный процесс; R – отношения, установленные для субъектов учебного процесса (горизонтальные: лектор-ассистент, лектор-диспетчер; вертикальные: студент-лектор, студент-ассистент, студент-диспетчер); Act – множество действий, выполняемых субъектами; I – множество установленных видов общения и взаимодействий между субъектами; T – расписание учебных занятий; U – состояние выполнения учебного плана студентом (экзаменационные ведомости, учебно-методические карты дисциплин).

Учебный процесс на кафедре (K) выполняется посредством действий (Act) и взаимодействий (I) между конкретными субъектами (X_i, L_i, P_i, Y_i), определяемыми кафедрой отношением R (например, лектор-дисциплина-студент, лектор-ассистент), происходящих по строгому расписанию (T). Кафедра, как среда для проведения учебного процесса, может накладывать различные ограничения, например, в виде дефицита аудиторий. Таким образом, учебный процесс как объект моделирования является распределенным и динамичным.

Ставится цель создать такую компьютерную среду обучения (виртуальную кафедру K'), в которой сохраняются все необходимые для учёбы отношения (R) и устраняются жесткие пространственно-временные ограничения в виде расписания занятий (T) (рис. 1):

$$УП = \{S, K', R, Act, I, U\}.$$

Такая среда может быть построена только на основе агентно-ориентированного подхода, использующего принципы распределенного искусственного интеллекта [1].

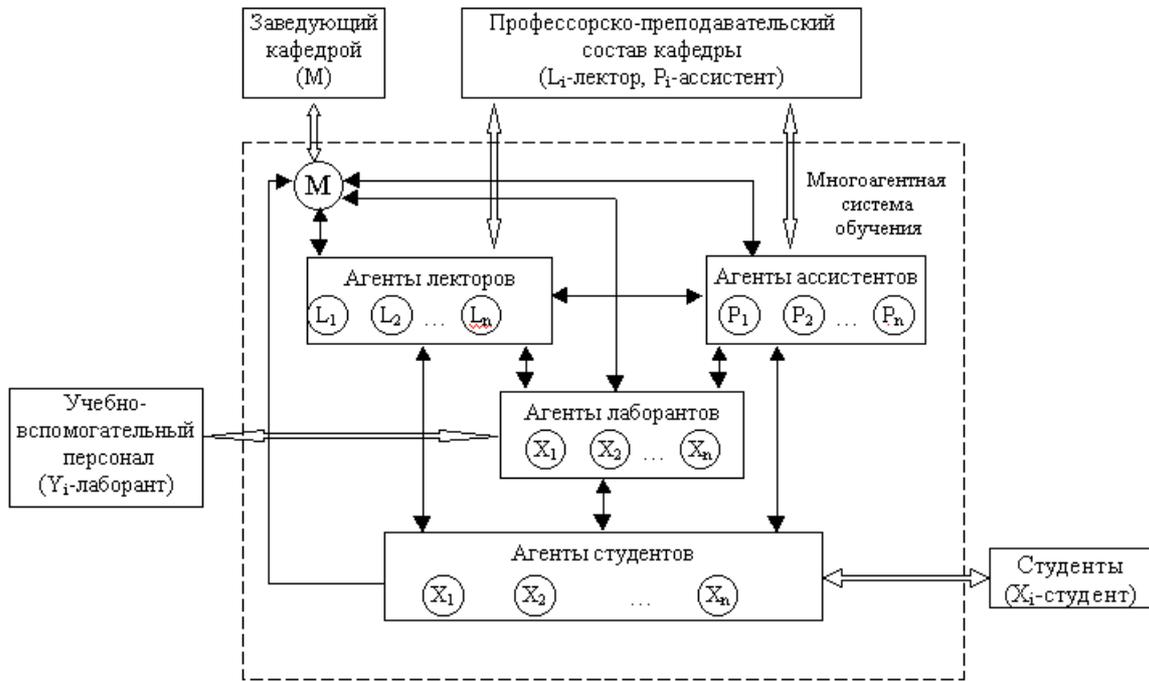


Рис. 1. Схема организации индивидуального обучения студентов на основе агентно-ориентированного подхода

Существующие в настоящее время обучающие системы предоставляют обучающимся связанный гипертекстовыми ссылками мультимедийный учебный материал, осуществляют контроль знаний по изучаемому предмету, обеспечивают асинхронный и индивидуальный режим работы, но при этом не поддерживают в достаточной мере необходимый при обучении диалог "преподаватель-студент", что снижает качество и оперативность процесса обучения.

2. Многоагентная модель распределенной системы

В настоящее время в качестве объектов анализа, проектирования и управления все чаще рассматриваются сложные распределенные системы (подразделения промышленных предприятий, финансовые организации, производственно-коммерческие фирмы, учебные заведения и т.д.). Компоненты распределенной системы могут обладать сложным поведением и индивидуальными средствами коммуникации, характерными для человека или трудно формализуемого объекта, что делает невозможным применение известных математических методов для их описания. Такие сложные системы, обладающие распределенным интеллектом, могут быть адекватно описаны методами теории многоагентных систем. Многоагентная система (МАС) строится как

объединение отдельных подсистем (агентов), основанных на знаниях, и формально определяется следующим образом [1]:

$$MAC = (A, E, R, ORG, ACT, COM, EV),$$

где A – множество агентов, способных функционировать в некоторых средах E , находящихся в определенных отношениях R и взаимодействующих друг с другом, формируя некоторую организацию ORG , обладающих набором индивидуальных и совместных действий ACT (стратегий поведения и поступков), включая возможные коммуникативные действия COM , и характеризуется возможностями эволюции EV .

Основное свойство агентов связано с его автономностью, т.е. способностью функционировать без вмешательства человека и при этом осуществлять самоконтроль над своими действиями и внутренним состоянием [1, 2]. В этом состоит принципиальное отличие агента от понятия объекта в ООП.

Технология MAC широко применяется для создания моделей функционирования сложных объектов, имеющих распределенную структуру. Агентно-ориентированный подход использован для построения интеллектуальной системы управления целевыми программами [2], системы управления воздушным движением [3], среды интерактивного обучения курсантов военного вуза тактике ведения боевых действий [4]. На базе сети интеллектуальных агентов разработан сервер дистанционного обучения [5]. Для решения проблем, связанных с разработкой интерактивного общения преподавателя и студента, авторы [6] считают перспективным использование многоагентных систем, так как с их помощью осуществляется переход от локализованного к распределенному искусственному интеллекту. В работе [7] агентный подход применен к созданию компьютерной системы поддержки дистанционного обучения. Прототип такой системы создан с применением инструментария разработки многоагентных систем $JADE$.

Агентно-ориентированный подход позволяет также снизить сложность создания программных систем, гарантировать их надежность и упростить их сопровождения.

3. Методологии и средства реализации MAC

Для разработки MAC созданы методологии и на их основе различные технологии, являющиеся развитием существующих методов инженерии знаний и объектно-ориентированного программирования. К объектно-ориентированному классу относится технология моделирования систем на основе BDI агентов [3]. В сфере совместной работы применяется многоагентный метод *Multi-Agent Scenario-Based Method (MASB)*, основанный на сценарии [8]. Для моделирования промышленных предприятий предложена специализированная агентно-ориентированная

методология, объединяющая объектно ориентированный метод Object Oriented Software Engineering (OOSE) и метод моделирования предприятий IDEF и CIMOSA [9]. Методология Gaia применяется для агентно-ориентированного анализа и проектирования MAC, разрабатываемых для предметных областей широкого назначения [10]. Позже методология Gaia была доработана с тем, чтобы сделать ее применимой для разработки агентно-ориентированных Internet приложений [11].

Распределенное решение задачи сообществом автономных агентов, обладающих реактивными, когнитивными, кооперативными и социальными знаниями, получено с помощью метода CoMoMAS [12], который является развитием метода CommonKADS [13]. Метод MAS-CommonKADS [14] также является расширением метода CommonKADS и поддерживается объектно-ориентированными технологиями. К ним относятся технология моделирования объектов (Object Modelling Technique - OMT), технология объектно-ориентированной разработки программного обеспечения (Object Oriented Software Engineering - OOSE), технология проектирования на основе обязательств (Responsibility Driving Design - RDD), технологии описания протоколов агентов (SDL и MSC96).

Для анализа предметной области и проектирования MAC применяются инструментальные средства AgentBuilder и JADE, который генерируют приложения на языке Java.

4. Многоагентная модель интерактивного обучения студентов

Для разработки агентно-ориентированной модели процесса обучения на кафедральном уровне применялась универсальная методология Gaia [10], позволяющая учитывать особенности системы обучения на макро и микро уровнях. С помощью этой методологии выполнен агентно-ориентированный анализ обучения, который позволил системно перейти от этапа постановки задачи на создание компьютерной среды обучения к этапу проектирования различных моделей виртуальной кафедры, удобных для программной реализации.

Модели процесса обучения и взаимосвязи между ними, формируемые по методологии Gaia, представлены на рис. 2.

Для детализации моделей выделены абстрактные и конкретные понятия поставленной задачи. К абстрактным понятиям отнесены роли, полномочия, обязательства, протоколы, активность, обязательства жизнеспособности, условия безопасности. Эти понятия используются во время анализа для концептуализации системы. Конкретные понятия (например, агентные типы, услуги, связи между агентами) используются на этапе проектирования и имеют непосредственный программный аналог в системе.

МАС обучения на уровне кафедры рассматривается как набор ролей, которые определенным образом взаимосвязаны между собой и принимают участие в постоянных и наделённых законным статусом образцах взаимодействия с другими ролями.

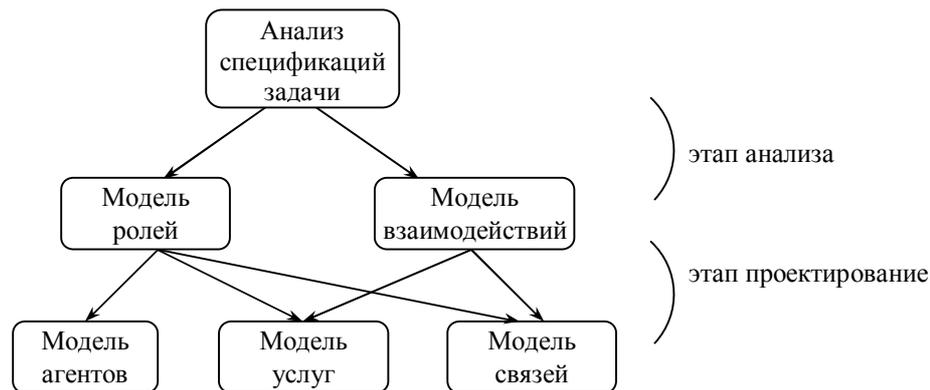


Рис. 2. Взаимосвязи между моделями методологии Gaia

Роль рассматривается как абстрактное описание ожидаемых функций должностного лица, поэтому в МАС обучения по дисциплине выделены роли студента, лектора, лаборанта и других сотрудников кафедры. Роли характеризуются полномочиями и обязательствами. Полномочия устанавливают ресурсы, которые могут обоснованно использоваться для выполнения роли, возможность роли генерировать информацию, только чтение информации или возможность ее изменять. В учебном процессе агент студента должен быть в состоянии получать учебный материал, генерировать информацию, содержащую ответы на тестовые задания или экзаменационный билет. Агенту лектора необходимо вносить изменения в конспект лекций, журнал выполнения учебного плана студентами, составлять конспект лекций, тестовые задания, экзаменационные билеты, выставлять оценки, получать ответы студентов.

Полномочия роли специфицируются ключевыми словами **read** (доступ к информации без возможности ее изменять), **changes** (возможность изменять информацию) и **generates** (возможность генерировать информацию). Например, спецификация полномочий для роли студент показана на рис. 3.

Функции роли описываются в виде обязательств, которые делятся на две категории: жизнеспособность и условия безопасности. Обязательства жизнеспособности показывают, что будет сделано агентом и определяются

выражениями жизнеспособности, которые определяют "жизненный цикл роли". Выражения жизнеспособности являются регулярными выражениями. Выражения жизнеспособности устанавливают возможность выполнения траектории "жизненного цикла" посредством деятельности и взаимодействий, связанных с ролью. Деятельность – это то, что агент может выполнить, не прибегая к взаимодействию с другими агентами, а взаимодействия с другими ролями определяются протоколами.

generates	данные студента
generates	запрос учебного материала
reads	УМКД
reads	зачетная книжка
reads	конспект лекций
reads	тестовое задание
generates	ответ на тестовое задание
reads	экзаменационный билет
generates	ответ на экзаменационный билет
reads	экзаменационная оценка
reads	журнал успеваемости (выполнение учебного плана студентом)

Рис. 3. Спецификация полномочий для роли СТУДЕНТ

Выражения жизнеспособности для роли СТУДЕНТ показаны на рис.4.

<p>СТУДЕНТ = ЗАРЕГИСТРИРОВАТЬСЯ ЗАРЕГИСТРИРОВАТЬСЯ . (Сеанс обучения лекц материалу Экзамен) . ЗАВЕРШИТЬ СЕАНС Сеанс обучения лекц материалу = (ЗАПРОСИТЬ УЧЕБНЫЙ МАТЕРИАЛ, ((СЛУШАТЬ ЛЕКЦИЮ ПЕРЕРВАТЬ ЗАНЯТИЕ) (ОТВЕТИТЬ НА ТЕСТ ВОПРОС ПОВТОРИТЬ ПРОЙДЕННЫЙ МАТЕРИАЛ ПЕРЕРВАТЬ ЗАНЯТИЕ)⁺ . ОТПРАВИТЬ ВЫПОЛНЕННОЕ ЗАДАНИЕ НА ПРОВЕРКУ)[*] Экзамен = ЗАПРОСИТЬ УЧЕБНЫЙ МАТЕРИАЛ, ОТВЕТИТЬ НА ЭКЗ. БИЛЕТ, ПЕРЕРВАТЬ ЗАНЯТИЕ, ОТПРАВИТЬ ВЫПОЛНЕННОЕ ЗАДАНИЕ НА ПРОВЕРКУ</p>

Рис. 4. Регулярные выражения жизнеспособности для роли СТУДЕНТ

Условия безопасности в методологии Gaia специфицируются посредством предикатов. В предикатах используются переменные, обозначающие атрибуты полномочий роли. Например, для выполнения роли СТУДЕНТ необходимо, чтобы после ввода пароля студент был идентифицирован.

Полное описание роли СТУДЕНТ показано на рис. 5.

Аналогичным образом определены схемы других ролей, участвующих в учебном процессе на кафедральном уровне. Модель ролей (рис. 2) определяется всей совокупностью схем ролей процесса.

Динамика МАС определяется взаимодействием между ролями, поэтому на следующем этапе анализа рассматривались основные виды и назначения взаимодействий. Совокупность взаимодействий между ролями образуют модель взаимодействий (рис. 2), которая состоит из набора установленных протоколов. Для каждого вида взаимодействия

составляется отдельный протокол. Общая структура протокола показана на рис. 6.

Схема роли СТУДЕНТ	
Описание роли Роль СТУДЕНТ предназначена для обучения студента конкретной учебной дисциплине. Обучение состоит из прослушивания лекций, выполнения тестовых заданий (для проверки усвояемости лекционного материала), выполнения практических заданий, сдачи экзамена. Студент запускает приложение, выполняет авторизацию, после чего загружается его интерфейс.	
Протоколы и Активность ЗАРЕГИСТРИРОВАТЬСЯ ЗАПРОСИТЬ УЧЕБНЫЙ МАТЕРИАЛ <u>СЛУШАТЬ ЛЕКЦИЮ</u> ПРЕРВАТЬ ЗАНЯТИЕ <u>ОТВЕТИТЬ НА ТЕСТОВЫЙ ВОПРОС</u> <u>ПОВТОРИТЬ ПРОЙДЕННЫЙ МАТЕРИАЛ</u> ОТПРАВИТЬ ВЫПОЛНЕННОЕ ЗАДАНИЕ НА ПРОВЕРКУ / СООБЩИТЬ ОБ ОКОНЧАНИИ ЛЕКЦИИ ЗАВЕРШИТЬ СЕАНС <u>ОТВЕТИТЬ НА ЭКЗ. БИЛЕТ</u>	
Полномочия generates данные студента generates запрос учебного материала reads УМКД reads зачетная книжка reads конспект лекций reads тестовое задание generates ответ на тестовое задание reads экзаменационный билет generates ответ на экзаменационный билет reads экзаменационная оценка reads журнал успеваемости (выполнение учебного плана студентом)	
Обязательства жизнеспособности СТУДЕНТ = ЗАРЕГИСТРИРОВАТЬСЯ ЗАРЕГИСТРИРОВАТЬСЯ . (Сеанс обучения лекц материалу Экзамен) . ЗАВЕРШИТЬ СЕАНС Сеанс обучения лекц материалу = (ЗАПРОСИТЬ УЧЕБНЫЙ МАТЕРИАЛ, ((СЛУШАТЬ ЛЕКЦИЮ ЗАКОНЧИТЬ ЗАНЯТИЕ) (ОТВЕТИТЬ НА ТЕСТ ВОПРОС ПОВТОРИТЬ ПРОЙДЕННЫЙ МАТЕРИАЛ ЗАКОНЧИТЬ ЗАНЯТИЕ) ⁺ . ОТПРАВИТЬ ВЫПОЛНЕННОЕ ЗАДАНИЕ НА ПРОВЕРКУ) [*] Экзамен = ЗАПРОСИТЬ УЧЕБНЫЙ МАТЕРИАЛ, <u>ОТВЕТИТЬ НА ЭКЗ. БИЛЕТ</u> , ПРЕРВАТЬ ЗАНЯТИЕ. ОТПРАВИТЬ ВЫПОЛНЕННОЕ ЗАДАНИЕ НА ПРОВЕРКУ условия безопасности <ul style="list-style-type: none"> • регистрация прошла успешно (получен допуск к работе) • допуск к экзамену – выполнен учебный план по дисциплине • допуск к тестированию – выполнен тест по предшествующей лекции 	

Рис. 5. Схема роли СТУДЕНТ

НАЗНАЧЕНИЕ ПРОТОКОЛА <i>краткое текстовое описание типа взаимодействия</i>		<i>информация, используемая ИНИЦИАТОРОМ для начала взаимодействия</i>
<i>ИНИЦИАТОР роль (роли), ответственные за начало взаимодействия</i>	<i>ОТВЕЧАЮЩИЙ роль (роли), с которой взаимодействует инициатор</i>	
Что выполняет ИНИЦИАТОР во время взаимодействия		<i>информация, доставляемая протоколу ОТВЕЧАЮЩЕГО во время взаимодействия</i>

Рис. 6. Общая структура протокола взаимодействия

На рис. 7 представлены примеры протоколов взаимодействий, связанных с ролью СТУДЕНТ.



Рис. 7. Примеры протоколов роли СТУДЕНТ: а) зарегистрироваться; б) запросить лекционный материал

К моделям нижнего уровня (см. рис. 2) относятся агентная модель, модель услуг и модель связей агентов. Модель агентов описывает используемые в системе агентные типы. Похожие роли (ЛЕКТОР, АССИСТЕНТ) объединены в один агентный тип Преподавателя. Агрегирование нескольких похожих ролей в один тип направлено на оптимизацию вычислительных ресурсов. Решение о совмещении нескольких ролей в один агентный тип не снижает понимание функционального назначения агентов, а только способствует повышению эффективности его программной реализации. Модель агентов системы представляется деревом агентных типов (рис. 8), в котором концевые вершины соответствуют ролям, определенным в модели ролей, а другие вершины соответствуют агентным типам.

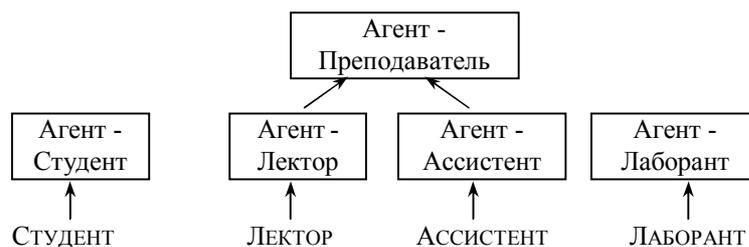


Рис. 8. Модель агентов МАС обучения студентов на кафедральном уровне

Модель услуг определяет функции, которые агент должен выполнять в соответствии с обязательствами жизнеспособности роли. Услуги можно сравнить с методами в объектно-ориентированном программировании, однако услуги не доступны для вызова другими агентами в отличие от методов объекта. Для каждой услуги, выполняемой агентом, в модели услуг определяются входные и выходные данные, пред и пост условия, в зависимости от соблюдения которых агент инициирует выполнение услуги

или услуга считается выполненной соответственно. Входные и выходные данные получают из активностей и протоколов взаимодействия роли. В табл. 2 собраны услуги, выполняемые агентом Студента.

Модель связей отражает возможные коммуникативные связи между агентами и представляется в виде ориентированного или неориентированного графа. Узлы соответствуют типам агентов, а дуги – коммуникационным путям. Каналы коммуникации между агентами предполагают передачу сообщений между агентами в обоих направлениях. На рис. 9 показана модель связей между тремя агентами, в которой каналы связи нагружены основными сообщениями. Модель связей построена на основе моделей ролей, взаимодействия и агентов.

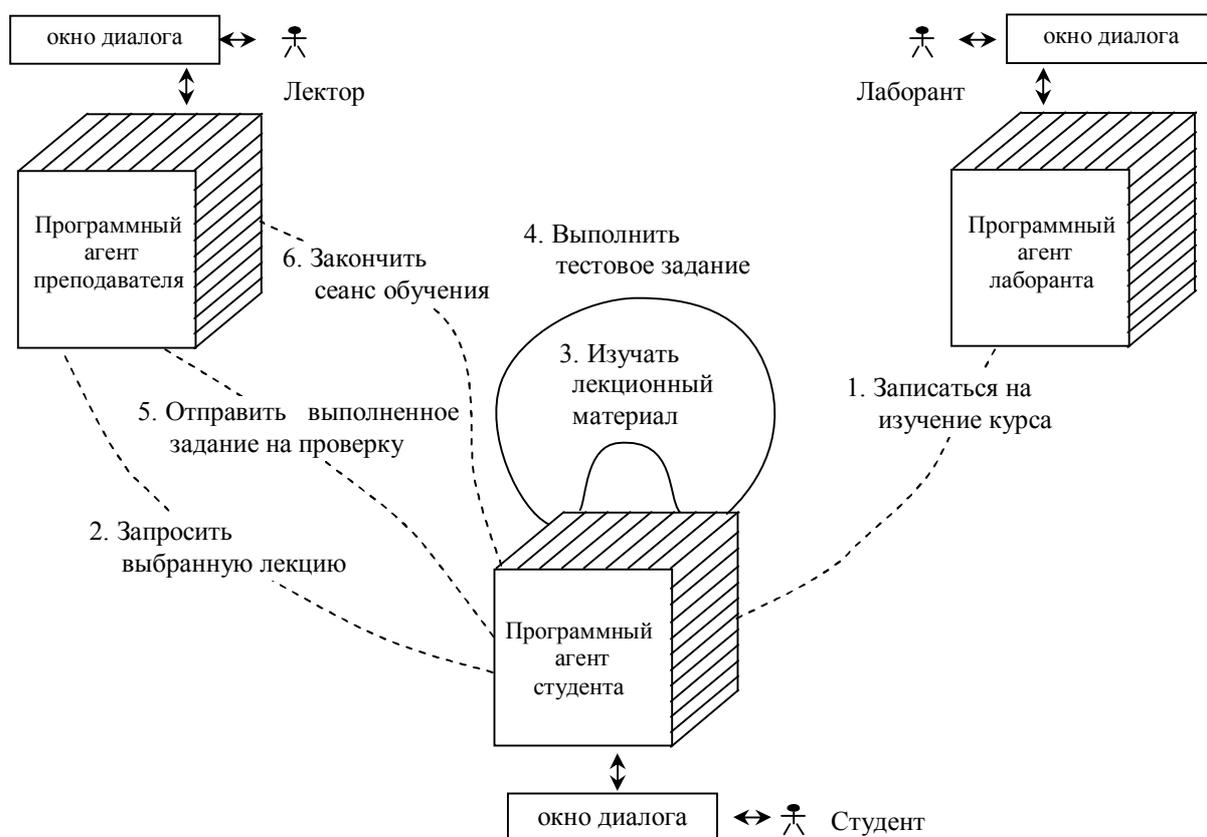


Рис. 9. Взаимодействие программных агентов при изучении студентом лекции без непосредственного участия сотрудников кафедры: (-----) – связь агентов по протоколу, (—) – активность агента, предусмотренная ролью.

Таблица 2. Модель услуг для агента Студент

Услуга	Входные данные	Выходные данные	Пред- условия	Пост- условия
Записаться на изучение дисциплины	ФИО, группа, название дисциплины	сообщение лаборанту	пользователь = студент	сообщение= отправлено
Запросить лекцию	ФИО, группа, название дисциплины № лекции	сообщение лектору	решение о допуске = допущен	сообщение = отправлено
Изучать лекционный	лекционный материал	вариант	лекция =	вариант завершения

материал		завершения лекции	предоставлена	лекции = {прослушана, прервана}
Запросить тестовое задание	ФИО, группа, название дисциплины № лекции	сообщение лектору		
Отвечает на вопросы тестового задания	тестовое задание	отчет по тестовому заданию, вариант завершения теста	вариант завершения лекции = прослушана	вариант завершения теста = {выполнено, прервано}
Отправить отчет лектору на проверку	отчет по тестовому заданию	сообщение лектору	вариант завершения теста = выполнено	сообщение = отправлено

Заключение

Выполнен агентно-ориентированный анализ процесса обучения студентов на кафедральном уровне, на основании которого получена новая модель индивидуального обучения студентов. Особенностью модели является сохранение взаимоотношений между участниками учебного процесса, близких к реально существующим, и предоставление возможности автономного и распределенного выполнения учебно-методических обязанностей. Эта модель позволяет повысить децентрализованность и индивидуальность работы всех участников процесса обучения на кафедральном уровне.

Перечень ссылок

1. Тарасов В.Б. От многоагентных систем к интеллектуальным организациям: философия, психология, информатика.- М.: “Едиториал УРСС”, 2002. -352 с.
2. Евгеньев Г.Б., Нехорошкин Н.И., Яценко В.В. Многоагентная система управления целевыми программами// Научная сессия МИФИ-2006. Сб. науч. тр. В 16 томах. Т.3. Интеллектуальные системы и технологии. – М.: МИФИ, 2006. – с. 26-27.
3. David Kinny, Michael Georgeff, Anand Rao A Methodology and Modelling Technique for Systems of BDI Agents. In W. van der Velde and J. Perram, editors, *Agents Breaking Away: Proceedings of the Seventh European Workshop on Modelling Autonomous Agents in a Multi-Agent MAAMAW'96, (LNAI Volume 1038)*. Springer-Verlag: Heidelberg, Germany, 1996.
4. Золотухин Ю.Н., Шидло Г.М. Подходы к реализации модели интеллектуальной системы интерактивного обучения тактике общевойскового боя // Сб. трудов X нац. конференции по искусств.

- интеллекту с междунар. участием КИИ-2006. – М.: Физматлит, 2006. – с.1085 - 1091.
5. Гаврилова Т.А., Яшин А.М., Фертман В.П. Взаимодействие интеллектуальных агентов для поддержки сервера дистанционного обучения// Материалы междунар. конф. "Интеллектуальные системы и информационные технологии в управлении IS&ITC", Псков, 2000, с.224-227.
 6. Курейчик В.М. Эволюционная адаптация интерактивных средств открытого образования /В.М. Курейчик, Л.А. Зинченко //Открытое образование. - 2001. - N1. - С.43-50.
 7. Глибовец Н.Н. Использование JADE (Java Agent Development Environment) для разработки компьютерных систем поддержки дистанционного обучения агентного типа// Educational Technology&Society 8(3) 2005 ISSN 1436-4522 pp. 325-345.
 8. Bernard Moulin and Mario Brassard. A scenario-based design method and an environment for the development of multiagent systems. In D. Lukose and C. Zhang, editors, *First Australian Workshop on Distributed Artificial Intelligence, (LNAI volumen 1087)*, pages 216–231. Springer-Verlag: Heidelberg, Germany, 1996.
 9. Elisabeth A. Kendall, Margaret T. Malkoun, and Chong Jiang. A methodology for developing agent based systems for enterprise integration. In D. Luckose and Zhang C., editors, *Proceedings of the First Australian Workshop on DAI*, Lecture Notes on Artificial Intelligence. Springer-Verlag: Heidelberg, Germany, 1996.
 10. Wooldridge M. J., Jennings N. R. and Kinny D. The Gaia methodology for agent-oriented analysis and design. *Autonomous Agents and Multi-Agent Systems*, 3(3):285–312, September 2000.
 11. Zambonelli F., Jennings N. R., Omicini A. and Wooldridge M. J. *Coordination of Internet Agents: Models, Technologies and Applications*, chapter 13. Springer-verlag, 2000. Agent-Oriented Software Engineering for Internet Applications.
 12. Norbert Glaser. *Contribution to Knowledge Modelling in a Multi-Agent Framework (the CoMoMAS M. P. Singh, and A. S. Rao, (Eds.), Springer-Verlag: Heidelberg, 1999.*
 13. A. Th. Schreiber, B. J. Wielinga, J. M. Akkermans, and W. Van de Velde. CommonKADS: A comprehensive methodology for KBS development. Deliverable DM1.2a KADSII/M1/RR/UvA/70/1.1, University of Amsterdam, Netherlands Energy Research Foundation ECN and Free University of Brussels, 1994.
 14. Carlos A. Iglesias, Mercedes Garijo, Jos'e C. Gonz'alez, and Juan R. Velasco. Analysis and design of multiagent systems using MAS-CommonKADS. In *AAAI'97 Workshop on Agent Theories, Architectures and Languages*, Providence, RI, July 1997. ATAL. (An extended version of

this paper has been published in *INTELLIGENT AGENTS IV: Agent Theories, Architectures, and Languages*, Springer Verlag, 1998.

Дата надходження до редакції 20.10.2006 р.