

УДК 004.896

## РАЗРАБОТКА АГЕНТНО-ОРИЕНТИРОВАННОЙ ИНТЕЛЛЕКТУАЛЬНОЙ СРЕДЫ ОБУЧЕНИЯ

Жабская Т.Е., Федяев О.И.

Донецкий национальный технический университет  
tanya\_zhabskaya@mail.ru

*В данной работе рассматривается создание виртуальной кафедры университета агентно-ориентированного типа. По методологии Gaia выполнен агентно-ориентированный анализ процесса обучения студентов. Для моделирования интеллектуальной деятельности преподавателей предлагается использование теории нечетких множеств.*

### **1. Введение**

Дистанционное обучение является областью активных исследований и развития. Интерес к исследованиям в данной области объясняется тем, что данный вид обучения является востребованным, т.к. позволяет учиться в соответствии с индивидуальными возможностями и желанием студентов в освоении дисциплин в своем собственном темпе. Однако, данный вид обучения обладает также недостатками, самым значительным из которых является отсутствие личного общения с преподавателями. Общение студентов с преподавателем является наиболее важной частью обучения, поскольку без общения с преподавателем, несмотря на полноценное учебно-методическое обеспечение курса, практически сложно получить хороший уровень знаний по предметам. Это объясняется тем, что преподаватель может найти индивидуальный подход к каждому студенту на всех этапах передачи и контроля усвоения знаний. Поэтому, цель исследований в области дистанционного обучения состоит в развитии сетевых образовательных технологий для разработки обучающих систем, обладающих такими качествами как адаптивность и интеллектуальность. В работе [1] предоставлен обзор различных адаптивных и интеллектуальных технологий, появившихся в результате проведенных исследований. Новейшие работы в этой области совершенствуют технологии интеллектуальных обучающих систем и адаптивных систем гипермедиа.

В данной работе рассматривается построение модели учебного процесса на уровне кафедры университета как единой целостной

системы обучения. Кафедра, как объект моделирования, является распределенной системой, субъекты которой находятся в определенных взаимоотношениях друг с другом и выполняют определенную интеллектуальную деятельность.

Для анализа и проектирования сложных распределенных систем применяется агентно-ориентированный (АО) подход [2]. Данный подход позволяет наиболее адекватно описать учебный процесс на уровне кафедры университета в виде множества автономных взаимодействующих агентов, выполняющих делегированную им деятельность субъектов учебного процесса. Благодаря автономности агентов и гибким взаимодействиям между ними, данный подход позволяет справиться со сложностью создания программного обеспечения для крупных распределенных систем, гарантировать их качество и упростить их сопровождение.

АО подход для построения модели учебного процесса обладает потенциалом для реализации свойств адаптивности и интеллектуальности в среде дистанционного обучения посредством делегирования профессионального и педагогического опыта каждого преподавателя своему агенту. Также АО модель позволяет сохранить все необходимые для учёбы отношения и устранить жесткие пространственно-временные ограничения в виде расписания занятий, предоставить участникам образовательного процесса возможность автономного и дистанционного выполнения учебно-методических обязанностей. Для воспроизведения интеллектуальной деятельности преподавателей предлагается использование теории нечетких множеств [3].

## **2. Построение многоагентной модели учебного процесса**

Для разработки многоагентной системы (МАС), автоматизирующей процесс обучения на кафедральном уровне, был проведен АО анализ предметной области образовательного процесса по методологии Gaia [2].

Воспроизведение функций кафедры ВУЗа является сложной задачей, которую, в соответствии с методологией Gaia, естественно рассматривать как организацию множества действительно существующих и взаимодействующих ролей: Преподаватель, Студент, Лаборант. Роль является абстрактным описанием функций должностного лица и характеризуется самостоятельными действиями (активностями), взаимодействиями с другими ролями (общением), полномочиями в отношении необходимых для её выполнения ресурсов и обязательствами, которые определяют "жизненный цикл"

роли. Содержание ролей Студент и Преподаватель иллюстрируют рисунки 1а и 1б соответственно.

Схема роли Студент	Схема роли ПРЕПОДАВАТЕЛЬ
<b>Протоколы</b> Запрос лекции, Запрос теста по лекции, Предоставление теста, Ответы по тесту, Запрос результата тестирования, Запрос индивидуального задания, Запрос вопросов по индивидуальному заданию, Защита выполненного задания, Запрос на экзамен, Экзаменационный вопрос, Ответ на экзаменационный вопрос, Запрос результата экзамена	<b>Протоколы</b> = Запрос лекции, Предоставление лекции, Запрос теста по лекции, Сообщение о допуске к тесту, Предоставление теста, Запрос результата тестирования, Сообщение о результате тестирования, Запрос индивидуального задания, Предоставление индивидуального задания, Запрос вопросов, Предоставление вопросов, Защита выполненного задания, Результат защиты, Запрос на экзамен, Сообщение о допуске, Экзаменационный вопрос, Запрос результата экзамена, Сообщение результата экзамена
<b>Активности</b> <u>Изучать лекцию, Выполнить тест по лекции.</u> <u>Ответить на вопросы по индивидуальному заданию.</u> <u>Ответить на экзаменационный вопрос</u>	<b>Активности</b> = <u>Проверить допуск к тестированию, Проверить тест.</u> <u>Оценить уровень знаний, Проверить допуск к экзамену, Выставить оценку</u>
<b>Полномочия:</b> reads учебный план, reads лекции, reads тест по лекции, generates ответ на тест по лекции, reads результат теста, reads индивидуальное задание, reads список вопросов, generates ответы на вопросы по инд. заданию, reads оценка по инд. заданию, reads интегральная оценка, reads экзаменационный вопрос, generates ответ на экзаменационный вопрос, reads экзаменационная оценка, reads журнал успеваемости.	<b>Полномочия:</b> generates допуск к тестированию, generates тест по лекции, reads ответ на тест по лекции, generates результат теста, generates индивидуальное задание, generates список вопросов, reads ответы на вопросы по инд. заданию, generates оценка по инд. заданию, generates интегральная оценка, generates экзаменационный вопрос, reads ответ на экзаменационный вопрос, generates экзаменационная оценка, changes журнал успеваемости
<b>Обязанности: жизнеспособности</b> Студент = (Лекция   Практическое занятие   Экзамен) Лекция = (Изучение лекции   Выполнение теста) Изучение лекции = (Запросить лекцию, <u>Изучать лекцию</u> ) Выполнение теста = Запрос теста по лекции   (Запрос теста по лекции, Предоставление теста, <u>Выполнить тест по лекции</u> , Ответы по тесту, Запрос результата тестирования) Практическое занятие = Запрос индивидуального задания   Защитить выполненную работу Защитить выполненную работу = Запрос вопросов по индивидуальному заданию, <u>Ответить на вопросы по индивидуальному заданию</u> , Защита выполненного задания Экзамен = Запрос на экзамен   (Запрос на экзамен, (Экзаменационный вопрос, <u>Ответить на экзаменационный вопрос</u> ), Ответ на экзаменационный вопрос)+ Запрос результата экзамена) <b>условия безопасности</b> истина	<b>Обязанности: жизнеспособности</b> Преподаватель = (Лекция   Тестирование   Практическое занятие   Экзамен)* Лекция = Запрос лекции, Предоставление лекции Тестирование = (Запрос теста по лекции, <u>Проверить допуск к тестированию</u> , Сообщение о допуске к тесту)   (Запрос теста по лекции, <u>Проверить допуск к тестированию</u> , Сообщение о допуске к тесту, Предоставление теста, <u>Проверить тест</u> , Запрос результата тестирования, Сообщение о результате тестирования) Практическое занятие = Выдать задание   Задать вопросы   Оценить знания студента Выдать задание = Запрос индивидуального задания, Предоставление индивидуального задания Задать вопросы = Запрос вопросов, Предоставление вопросов Оценить знания = (Защита выполненного задания, <u>Оценить уровень знаний</u> , Результат защиты)   (Защита выполненного задания, <u>Оценить уровень знаний</u> , Результат защиты, <u>Выставить оценку</u> ) Экзамен = (Запрос на экзамен, <u>Проверить допуск к экзамену</u> , Сообщение о допуске)   (Запрос на экзамен, <u>Проверить допуск к экзамену</u> , Сообщение о допуске, (Экзаменационный вопрос)+ Запрос результата экзамена, Сообщение результата экзамена) <b>условия безопасности</b> previousLectureTest = false => testPermission = false; grade = failing => markInLog = nil, integralGrade = nil => permissionDecision = false

а)

б)

Рисунок 1 – Схемы описания ролей: а) роли Студент; б) роли Преподаватель.

На следующем этапе анализа рассматривались установленные для процесса обучения образцы взаимодействий. Для каждого вида общения составлен отдельный протокол, который определяет цель взаимодействия и необходимые данные. На рисунке 2 представлены образцы взаимодействий, установленные для общения преподавателей со студентами во время экзамена. Чтобы сдать экзамен, роль Студент взаимодействует с ролью Преподаватель по протоколу Запрос на экзамен (рис. 2а). Если преподаватель не допускает студента к экзамену, то выполняется только этот протокол. Если студент допущен к экзамену, то дальнейшее общение происходит по протоколам Экзаменационный вопрос (рис. 2б) и Запрос результата экзамена (рис. 2в).

Модель агентов предназначена для определения используемых в системе агентных типов и их количестве. В данной МАС каждой роли соответствует отдельный тип агента.

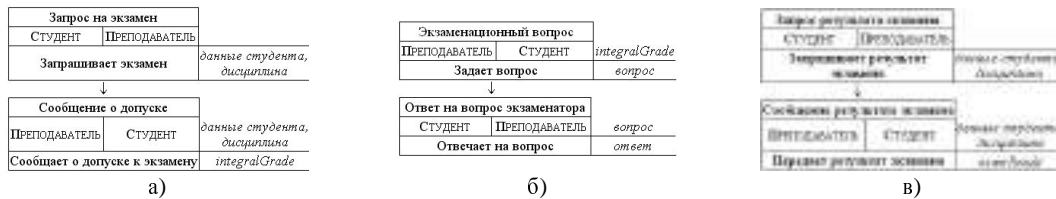


Рисунок 2 – Протоколы взаимодействия ролей СТУДЕНТ и ПРЕПОДАВАТЕЛЬ: а) ЗАПРОС НА ЭКЗАМЕН, б) ЭКЗАМЕНАЦИОННЫЙ ВОПРОС, в) ЗАПРОС РЕЗУЛЬТАТА ЭКЗАМЕНА.

В модели функционирования для каждого агента определены действия, которые должны выполняться этим агентом в соответствии с обязательствами жизнеспособности соответствующей роли. Для каждой функции, выполняемой агентом, в модели функционирования определены входные и выходные данные, пред- и постусловия. Входными и выходными данными являются данные обмена по протоколу между агентами. Пред- и постусловия представляют ограничения на выполнение функций и определяются в соответствии с условиями безопасности ролей агента. На рисунке 3 представлены функции агента ПРЕПОДАВАТЕЛЬ, идентифицированные на основании траектории “Экзамен” в выражении жизнеспособности роли ПРЕПОДАВАТЕЛЬ (рис. 1б). Функция “Проверить допуск к экзамену” проверяет, сдал ли студент все индивидуальные задания по данному предмету, и обрабатывает протокол ЗАПРОС НА ЭКЗАМЕН. Если не все задания сданы, то выполняется только первая ветвь выражения жизнеспособности “Экзамен”. Если все задания сданы, то выполняется вторая ветвь данного выражения. Входные данные – *данные студента* и *дисциплина*, выходные данные – *integralGrade*. Для данной функции пред- и постусловия отсутствуют. Функция “Провести экзамен” обрабатывает протокол ЭКЗАМЕНАЦИОННЫЙ ВОПРОС. Входные данные для данной функции – *ответ студента*, выходные данные – *экзаменационный вопрос*. С данной функцией связано предусловие  $integralGrade \neq nil$ , постусловие – допустимые значения *examGrade*. Функция “Сообщить результат экзамена” обрабатывает протокол ЗАПРОС РЕЗУЛЬТАТА ЭКЗАМЕНА. Входные данные для данной функции – *данные студента* и *дисциплина*, выходные – *examGrade*.

Функции	Вх. данные	Вых. данные	Предусловие	Постусловие
проверить допуск студента к экзамену	<i>данные студента, дисциплина</i>	<i>integralGrade</i>	истина	истина
провести экзамен	<i>ответ студента</i>	<i>экзаменационный вопрос</i>	$integralGrade \neq nil$	<i>результат экзамена</i> ∈ {"отл", "хор", "удов", "неуд"}
сообщить результат экзамена	<i>данные студента, дисциплина</i>	<i>результат экзамена</i>	истина	истина

Рисунок 3 – Функции агента ПРЕПОДАВАТЕЛЬ для проведения экзамена.

В моделі зв'язей отражені можливі комунікативні зв'язі між агентами. Ця модель сформована на основі моделі ролей і моделі взаємодії. На її основі для кожного агента складено список агентів, з якими можливо взаємодія.

Розроблені АО моделі представляють собою концептуальне описання діяльності суб'єктів навчального процесу, яку далі необхідно делегувати їх програмним агентам. В роботі [4] запропонована методика переходу від концептуальних моделей Gaia в фізичні моделі інструментаріа JACK, заснованого на BDI архітектурі агента.

### 3. Моделювання інтелектуальності

В процесі навчання важливою задачею викладача є об'єктивне оцінювання знань, отриманих студентами. Функція агента ПЕРЕДАВАТЕЛЬ “провести екзамен” (рис. 3) реалізує нечіткий механізм оцінки знань на основі теорії нечітких множин Л.Заде[3]. В залежності від інтегральної оцінки (*integralGrade*) визначається попередній рівень знань студента: високий, хороший, задовільний. Викладач дисципліни розробляє набір питань для кожного рівня знань. В час екзамену агент ПЕРЕДАВАТЕЛЬ задає студенту питання, відповідні його поточному рівню знань. Поточний рівень знань визначається шляхом нечіткого логічного висновку за алгоритмом Мамдани [5] на основі фактів про попередні відповіді студента. Кожен раз агент ПЕРЕДАВАТЕЛЬ виконує переоцінку поточного рівня знань студента і, відповідно до результату переоцінки, задає наступне питання. По мірі збільшення правильних відповідей збільшується складність задаваних питань. Відповідно, по мірі збільшення неправильних відповідей складність задаваних питань зменшується. В разі впевненості агента ПЕРЕДАВАТЕЛЬ в високому рівні знань студента екзамен закінчується з відмінною оцінкою досить швидко. Дане поведіння агента ПЕРЕДАВАТЕЛЬ визначається нечіткою базою знань, що представляє собою сукупність продукційних правил (рис. 4).

- |  |   |
|--|---|
| 1. ЕСЛИ <i>ответ</i> = <i>правильный</i><br>ТО <i>текущий уровень знаний</i> = <i>растет</i>   | 7. ЕСЛИ <i>текущий уровень знаний</i> = <i>стабильно растет</i><br>И <i>уровень вопроса</i> = <i>средняя сложность</i><br>ТО <i>уровень вопроса</i> = <i>повышенная сложность</i> |
| 2. ЕСЛИ <i>ответ</i> = <i>близкий к правильному</i><br>ТО <i>текущий уровень знаний</i> = <i>не изменяется</i>   | 8. ЕСЛИ <i>текущий уровень знаний</i> = <i>стабильно растет</i><br>И <i>уровень вопроса</i> = <i>элементарный</i><br>ТО <i>уровень вопроса</i> = <i>средняя сложность</i>         |
| 3. ЕСЛИ <i>ответ</i> = <i>неточный</i><br>ТО <i>текущий уровень знаний</i> = <i>понижается</i>   | 9. ЕСЛИ <i>текущий уровень знаний</i> = <i>падает</i><br>И <i>уровень вопроса</i> = <i>повышенная сложность</i><br>ТО <i>уровень вопроса</i> = <i>средняя сложность</i>           |
| 4. ЕСЛИ <i>ответ</i> = <i>не правильный</i><br>ТО <i>текущий уровень знаний</i> = <i>падает</i>  | 10. ЕСЛИ <i>текущий уровень знаний</i> = <i>падает</i><br>И <i>уровень вопроса</i> = <i>средняя сложность</i><br>ТО <i>уровень вопроса</i> = <i>элементарный</i>                  |
| 5. ЕСЛИ <i>текущий уровень знаний</i> = <i>стабильно растет</i><br>И <i>уровень вопроса</i> = <i>повышенная сложность</i><br>ТО <i>экзаменационная оценка</i> = <i>отлично</i> | 11. ЕСЛИ <i>текущий уровень знаний</i> = <i>падает</i><br>И <i>уровень вопроса</i> = <i>элементарный</i><br>ТО <i>уровень вопроса</i> = <i>элементарный</i>                       |
| 6. ЕСЛИ <i>текущий уровень знаний</i> = <i>растет</i><br>И <i>уровень вопроса</i> = <i>повышенная сложность</i><br>ТО <i>уровень вопроса</i> = <i>повышенная сложность</i>     |   |

Рисунок 4 – Фрагмент нечіткої бази знань.

Антецеденты *ответ* и *текущий уровень знаний*, используемые в вышеприведенных продукционных правилах являются лингвистическими переменными (ЛП). Множество значений ЛП *ответ* определяется множеством термов: {правильный, близкий к правильному, неточный, неправильный}. Данное терм-множество характеризуется функциями принадлежности, показанными на рис. 5а. Множество значений ЛП *текущий уровень знаний* определяется терм-множеством: {падает, понижается, растет, повышается, стабильно растет, стабильно падает}. Функции принадлежности ЛП данному терм-множеству показаны на рис. 5б и 5в.

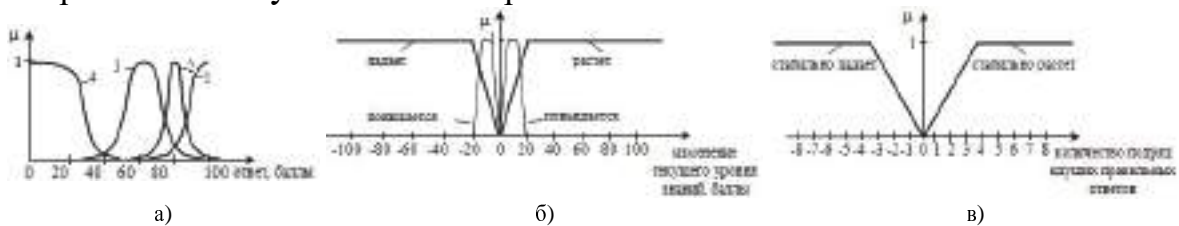


Рисунок 5 – Функции принадлежности нечетким множествам: а) 1- правильный, 2- близкий к правильному, 3 – неточный, 4 – ошибочный; б) падает, понижается, повышается, растет; в) стабильно падает и стабильно растет.

Интеллектуальность заключается в способности агента Преподавателя адаптироваться к индивидуальным возможностям каждого студента к усвоению дисциплины и оценивать глубину его понимания предмета.

#### 4. Заключение

Построена многоагентная модель кафедры университета, которая способна более реалистично описать трудно формализуемый учебный процесс благодаря BDI-архитектуре искусственных агентов и возможности использования различных видов взаимодействий между агентами. Использование BDI-архитектуры и нечётких систем позволили описать цели, обязательства, желания и намерения всех созданных искусственных агентов образовательного процесса, что в целом обеспечили интеллектуальность, адаптивность и гибкость их поведения. АО модель учебного процесса кафедры и средства визуализации позволяют оценивать в динамике различные ситуации, события и действия, связанные с передачей и усвоением учебной информации.

### Список литературы

1. Brusilovsky P., 1999. Adaptive and Intelligent Technologies for Web-based Education. In C. Rollinger and C. Peylo (eds.), Special Issue on Intelligent Systems and Teleteaching, Künstliche Intelligenz, 4, pp 19-25.
2. Zambonelli F., Jennings N. R., Wooldridge M., 2003. Developing Multiagent Systems: The Gaia Methodology. In ACM Transactions on Software Engineering Methodology, 12(3):317-370.
3. Zadeh L.A., 1975. The concept of a linguistic variable and its application to approximate reasoning. Information Sciences, 1:119-249.
4. Федяев О.И., Жабская Т.Е. Семантика перехода от концептуальных к физическим моделям среды JАСК при создании многоагентных систем // Труды двенадцатой национальной конференции по искусственному интеллекту с международным участием КИИ-2010. Том 3. – М.: Физматлит, 2010. – С. 61-69.
5. Mamdani E.H., 1977. Applications of fuzzy set theory to control system: A survey Fuzzy Automata and Decision Processes / M.M. Gupta, G.H. Saridis and B.R. Gaines, eds. – New York: North – Holland.

Получено 08.09.2011