

Архитектура и функционирование программных агентов в обучающей системе многоагентного типа

Федяев О. И., Жабская Т. Е.

Донецкий национальный технический университет
fedyaev@r5.dgtu.donetsk.ua

Abstract

Fedyaev O., Zhabskaya T. Program agent architecture and functioning in multi-agent learning system. The submitted article considers agent-oriented models of the students' subject process studies, agents' intelligent architecture and agents' interaction organization in the tutorial system. The suggested MAS architecture will provide autonomy, individuality and state of distribution carrying out educational-methodological duties of all the education process participants at the department level.

Введение

Установленные в настоящее время формы получения высшего 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].

Учебный процесс формально может быть описан следующими компонентами:

$$УП = (S, K, R, Act, I, T, U),$$

где $S = \{M, (X_1, X_2, \dots, X_n), ((L_1, L_2, \dots, L_m), (P_1, P_2, \dots, P_k)), (Y_1, Y_2, \dots, Y_l)\}$ – 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).

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).

Для решения поставленной задачи предложена многоагентная модель виртуальной кафедры K' [1]:

$$УП = \{S, K', R, Act, I, U\}$$

$$K' = \{A, E, R, Act, Com, Org\},$$

где A – множество программных агентов субъектов учебного процесса, E – среда функционирования агентов, R – отношения между агентами, Act – набор индивидуальных и совместных действий агентов, Com – коммуникативные действия, Org – формируемая агентами организация.

Такая среда может быть построена на основе агентно-ориентированного подхода, использующего принципы распределенного искусственного интеллекта [1, 2, 3].

Агентно-ориентированное моделирование процесса изучения дисциплины

Для разработки многоагентного приложения, автоматизирующего процесс обучения на кафедральном уровне, был проведен агентно-ориентированный анализ предметной области образовательного процесса по методологии Gaia [4]. С помощью этой методологии разработаны модели, необходимые для описания виртуальной кафедры и последующей программной реализации.

Процесс обучения, в соответствии с методологией Gaia, описывается следующими моделями: моделью ролей, моделью взаимодействий, моделью агентов, моделью

услуг, моделью связей [5]. На рис. 1 показаны взаимосвязи и содержание моделей при агентно-

ориентированном

проектировании.

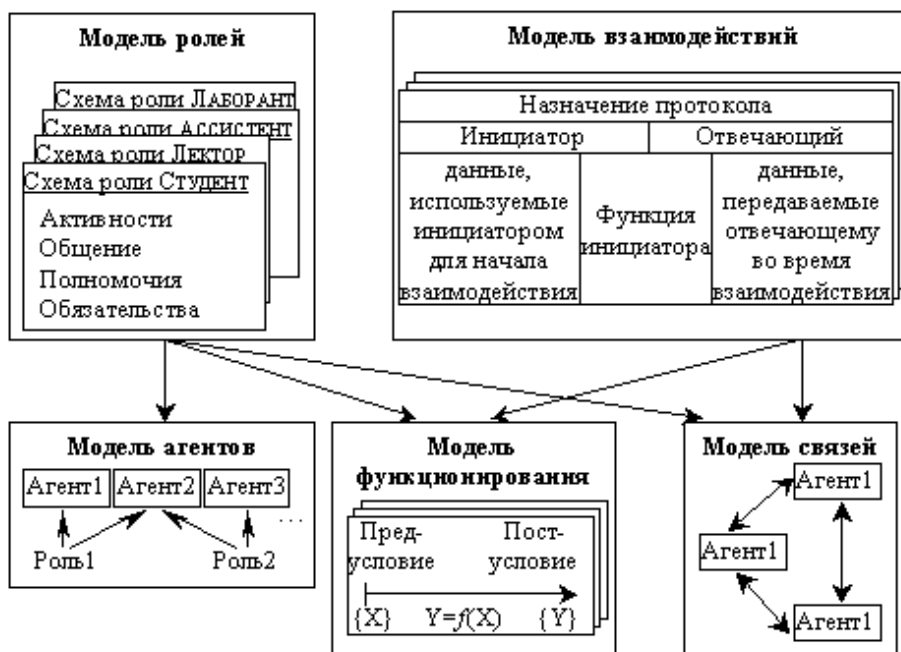


Рисунок 1 – Взаимосвязь моделей при агентно-ориентированном проектировании процесса обучения

Воспроизведение функций кафедры ВУЗа является сложной задачей, которую в соответствии с методологией Gaia, естественно рассматривать как организацию множества действительно существующих и взаимодействующих ролей: лектор, ассистент, лаборант, студент.

Поскольку роль является абстрактным описанием функций должностного лица, то каждая роль характеризуется самостоятельными действиями (активностями), взаимодействиями с другими ролями (общением), полномочиями в отношении необходимых для её выполнения ресурсов и обязательствами, которые определяют "жизненный цикл" роли и описываются регулярными выражениями [5]. Назначение модели ролей состоит в формальном описании содержания всех ролей организации.

На следующем этапе анализа рассматривались основные виды и назначения общения между ролями. Для каждого вида общения составлен отдельный протокол, который определяет последовательность действий ролей при взаимодействии и необходимые данные. Совокупность всех протоколов взаимодействий между ролями образует модель взаимодействий.

Модель агентов предназначена для определения используемых в системе агентных типов. Похожие роли объединены в один агентный тип. Решение о совмещении нескольких ролей в один агентный тип улучшает понимание функционального назначения агентов и

способствует повышению эффективности их программной реализации. Модель агентов системы представляется деревом агентных типов (рис. 1), в котором концевые вершины соответствуют ролям, определенным в модели ролей, а другие вершины соответствуют агентным типам. Модель агентов дает представление о типах агентов и их количестве в многоагентной системе (МАС).

В модели функционирования для каждого агента определены действия, которые ими должны выполняться в соответствии с обязательствами жизнеспособности соответствующей роли. Для каждой функции, выполняемой агентом, в модели функционирования определены входные и выходные данные, пред- и постусловия, в зависимости от которых агент инициирует выполнение и определяет завершение функции. Входные и выходные данные для функции получают из активностей ролей. Для функций, реализующих протоколы взаимодействия, они определены данными обмена между агентами.

В модели связей отражены возможные коммуникативные связи между агентами. Она сформирована на основе модели ролей, модели взаимодействия и модели агентов (рис. 1). Модель связей представлена в виде ориентированного графа. Узлы соответствуют типам агентов, а дуги – коммуникационным связям. Связи коммуникации между агентами предполагают передачу сообщений между агентами в обоих

направлениях. На её основе для каждого агента составлен список агентов с которым возможно установление односторонних или двусторонних связей.

Разработанные агентно-ориентированные модели позволяют системно перейти от этапа постановки задачи к этапу программной реализации компьютерной среды с элементами квазиреального общения между субъектами учебного процесса изучения дисциплин кафедры.

Организация взаимодействия агентов

Поскольку учебный процесс основан на общении преподавателя со студентами, то работа МАС обучения построена на общении агентов [6]. Для каждого вида общения, определенного названием протокола в модели ролей, составлен протокол взаимодействия. Совокупность всех протоколов взаимодействия ролей составляет модель взаимодействий. На рис.2 представлен пример протокола взаимодействия между ролями СТУДЕНТ и ЛЕКТОР при запросе лекции. В начале агент Студент формирует запрос на лекцию и отправляет сообщение Лектору со своими данными и темой запрашиваемой лекции. Лектор принимает сообщение Студента и возвращает ему лекционный материал.



Рисунок 2 – Протокол запроса лекции на её изучение

Для реализации общения между агентами выбран язык коммуникации агентов KQML [7], который предлагает формат сообщений и протоколы управления сообщениями для совместного использования знаний интеллектуальными агентами во время выполнения МАС.

Язык KQML был разработан на основе теории речевых актов с учётом требований, предъявляемых к языкам коммуникации агентов. Его широкое применение обусловлено следующими причинами:

- синтаксис языка достаточно прост для грамматического разбора сообщений;
- разработаны типовые примитивы языка (перформативы), учитывающие основные акты взаимодействия агентов;

- язык обеспечивает общение между агентами в разнородной, распределенной и динамичной среде благодаря присвоению перформативам семантики сообщений.

Таким образом, чтобы общаться друг с другом, агенты должны уметь формировать запросы и ответы в формате сообщений языка KQML. В соответствии с моделью взаимодействия, для каждого протокола взаимодействия ролей разработаны сообщения в

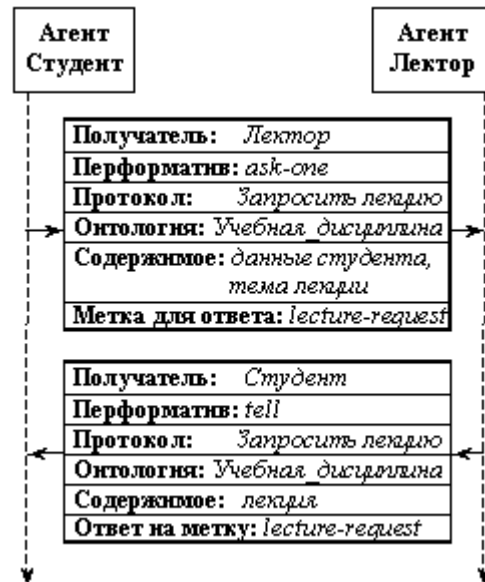


Рисунок 3 – Формат сообщений при коммуникации агентов

формате KQML. Например, при запросе Студентом лекции у Лектора общение осуществляется посредством показанных на рис. 3 сообщений.

Архитектура интеллектуального агента

Поскольку программным агентам делегируются выполнение полномочий субъектов образовательного процесса, то они должны имитировать разумные рассуждения, которые в определенной степени соответствуют их профессиональной деятельности. Для имитации профессиональной деятельности каждый программный агент должен обладать знаниями о порученных должностных обязанностях, знаниями об агентах, с которыми возможно общение, а также правилами, определяющими его поведение в плане выполнения своих обязанностей. Из характера образовательного процесса следует, что кроме реактивности, автономности, активности и коммуникабельности, архитектура программного агента должна иметь внутренние механизмы мотивации, которые задаются ментальными свойствами, такими как убеждения, обязательства, способности и правила поведения.

В связи с этим, задача выбора адекватной архитектуры агентов рассматривалась с учётом перечисленных требований и содержания ролей, возложенных на программных агентов. Из существующей классификации для создаваемой

системы больше подходит архитектура, основанная на классических принципах искусственного интеллекта, т. е. архитектура интеллектуального агента на основе продукционных правил (рис. 4).

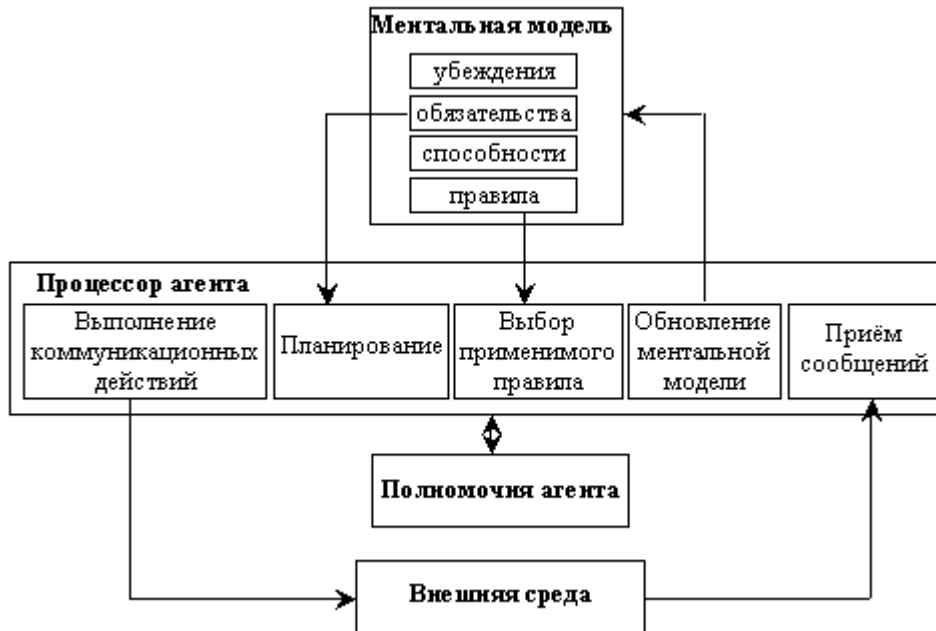


Рисунок 4 – Архитектура интеллектуального агента

Ментальная модель обеспечивает агента знаниями о среде и способностью рассуждений, а полномочия дают ему способность действовать и взаимодействовать со своим окружением. Процессор агента выполняет следующие функции: в момент запуска агента на выполнение инициализирует его начальную ментальную модель, на протяжении всего жизненного цикла агента принимает входящие сообщения, определяет применимые правила поведения, выполняет указанные в соответствующем правиле действия, осуществляет планирование.

Главной составляющей ментальной модели агента являются убеждения. Убеждения представляют знания агента о текущем состоянии его внутреннего и внешнего мира и могут обновляться по мере поступления новой информации из внешней среды.

Следующей ментальной составляющей агента являются способности, которые используются агентом для связи действия с необходимыми для их выполнимости условиями (предусловиями). К необходимым условиям относятся условия, которые должны быть удовлетворены до выполнения действия. Например, предусловием к запросу лекционного материала является факт выбора студентом вида занятия (а именно - лекции) и темы лекции. Списком способностей агента определены действия, которые агент может выполнить при

условии, что удовлетворены необходимые условия. Способности агента неизменны и сохраняются на протяжении всего его жизненного цикла. Если происходят изменения в убеждениях агента, то это приводит к изменению значения истинности предусловий способностей, что повлияет на выполнение действий агента.

В обязательства включены соглашения о действиях, выполняемых для других агентов в установленное время. Обязательства агента могут изменяться в течение его жизненного цикла.

Очень важной составляющей ментальной модели являются правила поведения агента. Они построены на основе модели функционирования и представлены в форме продукционных правил WHEN-IF-THEN, определяющие поведение агента в различных ситуациях. Эти правила определяют реакцию агента на внешние события и обновления его собственной ментальной модели в течение его жизненного цикла. Первая часть правила WHEN идентифицирует новые события, которые происходят в среде агента и содержат условия приёма входящих сообщений, получаемых от других агентов. Вторая часть правила IF сравнивает текущую ментальную модель с условиями, которые требуются для того, чтобы правило было применимым. Конструкция IF – части позволяет сопоставлять убеждения, обязательства, способности и намерения. THEN-часть определяет действия и ментальные

изменения, выполняемые в ответ на принятое сообщение и состояния ментальной модели. На рис. 5 показан пример правила для отправления агентом Студент сообщения агенту Лектор о предоставлении лекции на изучение. В данном правиле отсутствуют условия в части WHEN поскольку оно не связано с приёмом входящего сообщения. Для того чтобы правило сработало,

агент Студент должен быть убеждённым в том, что студент выбрал лекцию как вид занятия и сделал выбор темы лекции. IF-часть сопоставляет его убеждения *вид_занятия* со значением “лекция” и *выбор_темы_лекции* со значением “сделан”. Если результат сопоставления истинен, то выполняются действия, определенные в части THEN. Эти действия формируют

```

NAME Запрос лекции
WHEN
IF (вид_занятия="лекция") AND (выбор_темы_лекции = сделан)
THEN
    Message.Получатель:= Лектор
    Message.Перформатив:= "ask-one"
    Message.Протокол:= "Запросить лекцию"
    Message.Онтология:= "Учебная_дисциплина"
    Message.Содержимое:= данные_студента, тема_лекции
    Message.Метка_для_ответа:= "lecture-request"
    SEND(Message)
  
```

Рисунок 5 –Правило поведения агента Студент при запросе лекции

соответствующее сообщение агенту Лектор, которое отправляется методом SEND().

На рис. 6 показано правило приёма лектором сообщения от студента на предоставление лекции. В этом правиле WHEN-часть определяет условия приёма входящего сообщения, которые определяются именем

перформатива, названием протокола, именем онтологии. Если результат сопоставления является истинным, то, согласно THEN-части правила, агент Лектор отмечает в журнале, что данный студент запросил лекцию по выбранной теме на изучение, формирует соответствующее сообщение студенту и отправляет его.

```

NAME Прием запроса на лекцию
WHEN
    (Message.Перформатив = "ask-one")
    (Message.Протокол = "Запросить лекцию")
    (Message.Онтология = "Учебная_дисциплина")
    (Message.Метка_для_ответа = "lecture-request")
IF
THEN
    ОТМЕТИТЬ_СТУДЕНТА_В_ЖУРНАЛЕ(данные_студента, тема_лекции)
    Message.Получатель:= Студент
    Message.Перформатив:= "tell"
    Message.Протокол:= "Запросить лекцию"
    Message.Онтология:= "Учебная_дисциплина"
    Message.Содержимое:= Лекционный материал
    Message.Метка_для_ответа:= "lecture-request"
    SEND(Message)
  
```

Рисунок 6 – Правило поведения агента Лектор при приёме запроса лекции от студента

На основе полномочий ролей разработано объектно-ориентированное представление знаний агента, связанных с его профессиональной деятельностью и умением оперировать этими знаниями. Например, в процессе изучения дисциплины 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.

Заключение

В работе предложены агентно-ориентированные модели процесса изучения студентами кафедральных дисциплин, архитектура интеллектуального агента, механизм

взаимодействия агентов на основе сообщений. Формализация процесса обучения, выполненная по агентно-ориентированной методологии Gaia, позволяет создать компьютерную систему обучения, которая на основе сохранения близких к реально существующим взаимоотношениям между участниками учебного процесса и предоставления им возможности автономного и распределенного выполнения учебно-методических обязанностей повышает децентрализованность и индивидуальность работы всех участников процесса обучения на уровне кафедры.

С целью придания агентам гибкости при их настройке на выполнение широкого спектра обусловленных задачей функций, предложенная архитектура агента разделена на статическую и динамическую части. Динамическая часть включает ментальность агента и его профессиональную ориентацию. На статическую часть, неизменную для всех агентов, возложена типовая функция управления процессом интерпретации продукционных правил поведения.

Общение агентов, представленное моделью взаимодействий, реализовано в формате KQML-сообщений, как общепринятого стандарта обмена данными и наиболее удобного механизма при использовании различных языков программирования и программных платформ.

В дальнейшем планируется программная реализация системы на языке Java. На основе созданной компьютерной системы обучения нового поколения агентно-ориентированного типа будет проведен анализ адекватности ментальных моделей участникам учебного процесса и получена оценка эффективности агентной архитектуры в целом.

Литература

1. Тарасов В.Б. От многоагентных систем к интеллектуальным организациям: философия, психология, информатика.- М.: "Едиториал УРСС", 2002. -352 с.
2. Тарасов В.Б. Агенты, многоагентные системы, виртуальные сообщества: стратегическое направление в информатике и искусственном интеллекте// Новости искусственного интеллекта. №2, 1998. с. 5 - 63.
3. Городецкий В.И., Грушинский М.С., Хабалов А.В. Многоагентные системы (обзор)// Новости искусственного интеллекта. №2, 1998. с. 64 - 116.
4. F.Zambonelli, N.R. Jennings, and M.Wooldridge. Developing Multiagent Systems: The Gaia Methodology. In ACM Transactions on Software Engineering Methodology, 12(3):317-370, July 2003.
5. Федяев О.И., Жабская Т.Е., Грач Е.Г. Многоагентная модель процесса обучения студентов на кафедральном уровне //Наукові праці Донецького національного технічного університету. Серія "Проблеми моделювання та автоматизації проектування динамічних систем". Випуск 5(116). – Донецьк: ДонНТУ, 2006.- с. 105 - 116.
6. Пономарева С.М., Хорошевский В.Ф., Методы и модели коммуникации в профессиональных группах и группах интеллектуальных агентов// Труды 4-го сборника Международного семинара по прикладной семиотике, семиотическому и интеллектуальному управлению ASC/IC'99. РАИИ, 1999. с. 20 – 34.
7. Tim Finin, Yannis Labrou and James Mayfield, KQML as an agent communication language, invited chapter in Jeff Bradshaw (Ed.), "Software Agents", MIT Press, Cambridge, (1997).