

УДК 519.62



ФУНКЦИОНИРОВАНИЕ ПРОГРАММНОГО АГЕНТА ИНТЕРАКТИВНОГО ВЗАИМОДЕЙСТВИЯ ПОЛЬЗОВАТЕЛЕЙ СИСТЕМЕ ДИСТАНЦИОННОГО ОБУЧЕНИЯ

С. А. Марьин¹, А. П. Мишин², В. Б. Репка³¹Харьковская государственная академия культуры, г. Харьков, Украина, maryin_servey@mail.ru²ХНУРЭ, г. Харьков, Украина, askell@rambler.ru³ХНУРЭ, г. Харьков, Украина, victoria_repka@kture.kharkov.ua

Рассмотрен подход использования агентных технологий в организации систем дистанционного обучения. На примере агента интерактивного взаимодействия участников процесса обучения изложены основные принципы создания и функционирования программных агентов в мультиагентной системе дистанционного образования. Приведены модели и алгоритмы функционирования и взаимодействия агента.

ДИСТАНЦИОННОЕ ОБУЧЕНИЕ, АГЕНТ, ЗНАНИЯ, ЦЕЛИ, ФАКТЫ, ОНТОЛОГИЧЕСКАЯ МОДЕЛЬ, VDI, JADEX, XML

Введение

Основываясь на возможных видах взаимодействия в системах дистанционного образования (СДО): координатор-группа; координатор-слушатель; слушатель-координатор; слушатель-слушатель; группа-слушатель; слушатель-группа, можно достичь желаемого результата — повышения эффективности обучения и вовлечения слушателей в активную самостоятельную деятельность в системе ДО. В принципе для большинства людей это наиболее оптимальный выход, но тут же мы сталкиваемся с проблемой, что не всегда у человека существует возможность спокойно обучаться из-за нехватки времени, места расположения учебного заведения и проблем, связанных с теми или иными факторами, которые ограничивают доступ человека к получению образования.

СДО использует всемирную паутину как средство интерактивного взаимодействия пользователей и реальных преподавателей, организующих процесс обучения, используя для этого программные средства, базу знаний и управляющие утилиты, организующие процесс обучения. Популярными в данное время стали агентные технологии.

В данной работе предлагается создать агента интерактивного взаимодействия пользователей системы дистанционного обучения как средство, выполняющее коммутацию пользователей (координатор-слушатель; слушатель-координатор) на основании набора логически заданных функций и рекомендуемых действий и в соответствии с поставленным запросом.

1. Концепция и общее понятие мультиагентной системы дистанционного обучения

Классический подход к процессу ДО — наличие структурированных, объединённых в единый информационно-образовательный «пакет данных», предоставляемый пользователям посредством каких-либо информационных носителей или сред-

ствами коммутации. Пакет содержит необходимые данные, непосредственно требующиеся для организации процесса обучения студента. На данном этапе развития информационных и телекоммуникационных технологий данный подход изжил себя. Основным недостатком такого подхода есть то, что весь процесс обучения организовывается самим обучаемым. Таким образом, целесообразно представить процесс образования как систему вида: «база данных», содержащая материалы обучения + набор утилит, непосредственно организующих процесс обучения. Для этого требуется первоначально определить функции и задачи СДО, наиболее точно имитирующие реальный процесс обучения, используя имеющиеся цели, задачи и базу знаний системы и максимально ограничивающие вмешательство в процесс обучения реального тьютора.

Решая вопросы «самостоятельности системы» с ограничением вмешательства тьютора, необходимо рассматривать наличие какого-то механизма, который будет проводить самоадаптацию в соответствии с поставленными целями и действиями пользователей. В качестве такого механизма может выступать совокупность взаимосвязанных агентов: пользователи взаимодействуют с системой через агентов, система взаимодействует с пользователями через агентов, и агенты могут взаимодействовать с агентами «советуясь» и «общаясь» друг с другом.

В данной работе для решения вышеуказанной проблемы предлагается рассмотреть агента как интеллектуальное средство, выполняющее интерактивное взаимодействие объектов на основании набора логически заданных функций и рекомендуемых действий в соответствии с поставленным запросом для повышения эффективности работы системы. Предлагается создать блок интерактивного взаимодействия системы ДО, обучаемого и тьютора посредством общения в виде передачи

текстовой информации, что позволит сократить или полностью исключить интервал адаптации студентов и преподавателей к возможностям новых средств обучения.

2. Идея разработки агента взаимодействия пользователей

Вопрос о том, как обеспечить пользователей связью при дистанционном обучении лежит на поверхности и ответ типа: используя различные средства коммутации (электронная почта, различные пейджинговые системы (ICQ, MSN и др.), текстовые и голосовые чаты, конференц-связь [1,2]) - слишком общий, поскольку не учитывает того множества факторов, которые в совокупности влияют на обучение: затрачиваемые ресурсы, интерактивность общения и синхронность функционирования в СДО.

Выявив наиболее приемлемые функции, такие как: текстовая передача данных; доставка сообщений в online/offline режимах пользователей; отсутствие установки специализированного программного обеспечения; запуск с серверной стороны; поддержка многопоточного использования системы пользователями, создадим средство интерактивного взаимодействия пользователей СДО в виде агента (рис. 1). Программный агент интерактивного взаимодействия пользователей СДО состоит из двух модулей: модуль коммутации и модуль поддержки пользователя в отсутствие тьютора (так называемый модуль чата). Модуль коммутации использует службу рассылки текстовых сообщений типа чат, устанавливая двухстороннюю связь пользователя системы и тьютора.

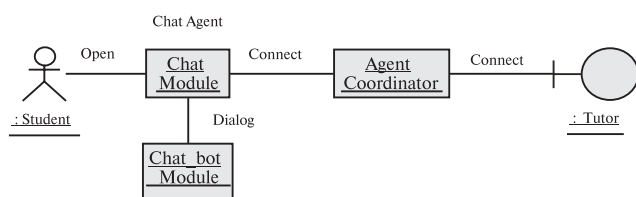


Рис. 1. Схема взаимодействия агентов с пользователями

Реагируя на заданный запрос, агент обрабатывает событие и принимает решение о связи с преподавателем; при отсутствии тьютора передаёт управление агенту, отвечающему за координацию взаимодействия в СДО. Модуль поддержки ожидает получения вопроса и выбора ответа на него. Модуль поддержки пользователя вырабатывает решение, базируясь на онтологической модели диалога.

Пользователи непосредственно взаимодействуют через агентов системы, не предпринимая никаких действий, кроме подачи запроса на установление возможности связи с участниками системы. Затем агент интерактивного взаимодействия

выполняет действия, связанные с построением диалогового окна общения пользователей и обработкой полученной информации об имеющихся пользователях. Эти действия выполняются по требованию каждого пользователя одними и теми же агентами [3].

Рассмотрим процесс совместного функционирования агентов и пользователя для более четкого определения рентабельности агента: при возникновении необходимости общения с тьютором в СДО для пользователей существует возможность запустить чат. Чат, связанный с модулем коммутации Start_Chat.agent, приводит в состояние активации класс-конструктор формирования: 1) интерфейса диалогового окна чата и 2) функцию связи с агентом координатором. Агент определяет состояние тьюторов, отправляя PING-запрос и формируя контакт-лист, после чего переходит в состояние ожидания дальнейших действий пользователя. «Пробуждение» модуля коммутации происходит в случае, если пользователь устанавливает связь с требуемым ему тьютором. Агент коммутации от лица обучаемого отправляет алерт-приглашение на коммутацию пользователей и устанавливает связь, запуская окно чата на стороне тьютора. Когда невозможно установить связь с тьютором, агент активизирует модуль Chat_bot, предоставляющий варианты выхода из ситуации, оставляя тьютору сообщение в offline-режиме для последующего ответа или связи со студентом.

3. Критерии функционирования агента Chat_Bot в системе ДО

При общении тьютора и студента диалог осуществляется по схеме вопрос-ответ. Рассмотрим все возможные варианты разговоров студента и тьютора для определения ключевых моментов диалога, которые не привязывают к конкретным вопросам и ответам, и для выявления закономерностей, дающих возможность применить их к определению задач агента Chat_Bot. Он является средством поддержки пользователей при отсутствии тьютора и выполняет роль преподавателя. Для формализации отношений понятий «Вопрос» - «Ответ» воспользуемся онтологиями [4], преимуществами которых для разрабатываемой агентной модели взаимодействия есть: однозначное определение терминов с использованием логических аксиом; логическая согласованность определений; монотонное расширение и/или специализации без ревизии существующих понятий; влияние кодирования. Концептуализация при создании онтологии специфицирована на уровне представления, а не символического кодирования. Для разработки онтологической модели O , воспользуемся следующей нотацией, — упорядоченная тройка:

$$O = (\mathcal{K}, \mathcal{R}, \Phi), \quad (1)$$

где \mathcal{K} – конечное множество концептов (понятий, терминов) предметной области, которую представляет онтология O ; \mathcal{R} – конечное множество отношений между концептами (понятиями, терминами) заданной предметной области; Φ – конечное множество функций интерпретации (аксиоматизации), заданных на концептах и/или отношениях онтологии O .

Рассмотрев отношение «вопрос пользователя – ответ бота» в виде онтологической модели, построенной на таксономической модели вариантов, классифицируем таксоны диалога, адаптируя их к среде ДО (рис. 2).

Первоначальное распознавание вопроса происходит в соответствии с представленной веткой таксономии Y_1 (вопрос пользователя) и продвигается по цепочке, присваивая идентификаторы вероятности совпадения вопроса с элементом дерева, такие как: $Y_{1,1}$ (тип) – первоначальное со-

стояние системного запроса, который определяет вопрос и относит его к одному из вариантов: требующему или не требующему ответа для возможного нормального функционирования системы и предотвращая ситуацию, при которой невозможно предоставить ответ пользователю. Если пользователь запрашивает у системы ответ на вопрос по несуществующей тематике, агент или игнорирует вопрос, или предлагает сформулировать его более точно, осуществляя запись во временный файл, содержащий неопределенные вопросы для дальнейшей обработки преподавателем или администратором системы. Когда установлен тип вопроса, система принимает соответствующее действие по обработке запроса и переходит на следующий шаг классификации вопроса к метке $Y_{1,2}$ (суть), выполняя алгоритм определения основных сущностей категории вопроса (схема алгоритма приведена на рис. 3). Раздел категорий подразумевает набор общепринятых ключевых слов, используемых в учебных курсах, таких как: знания, дополнения, уточнения.

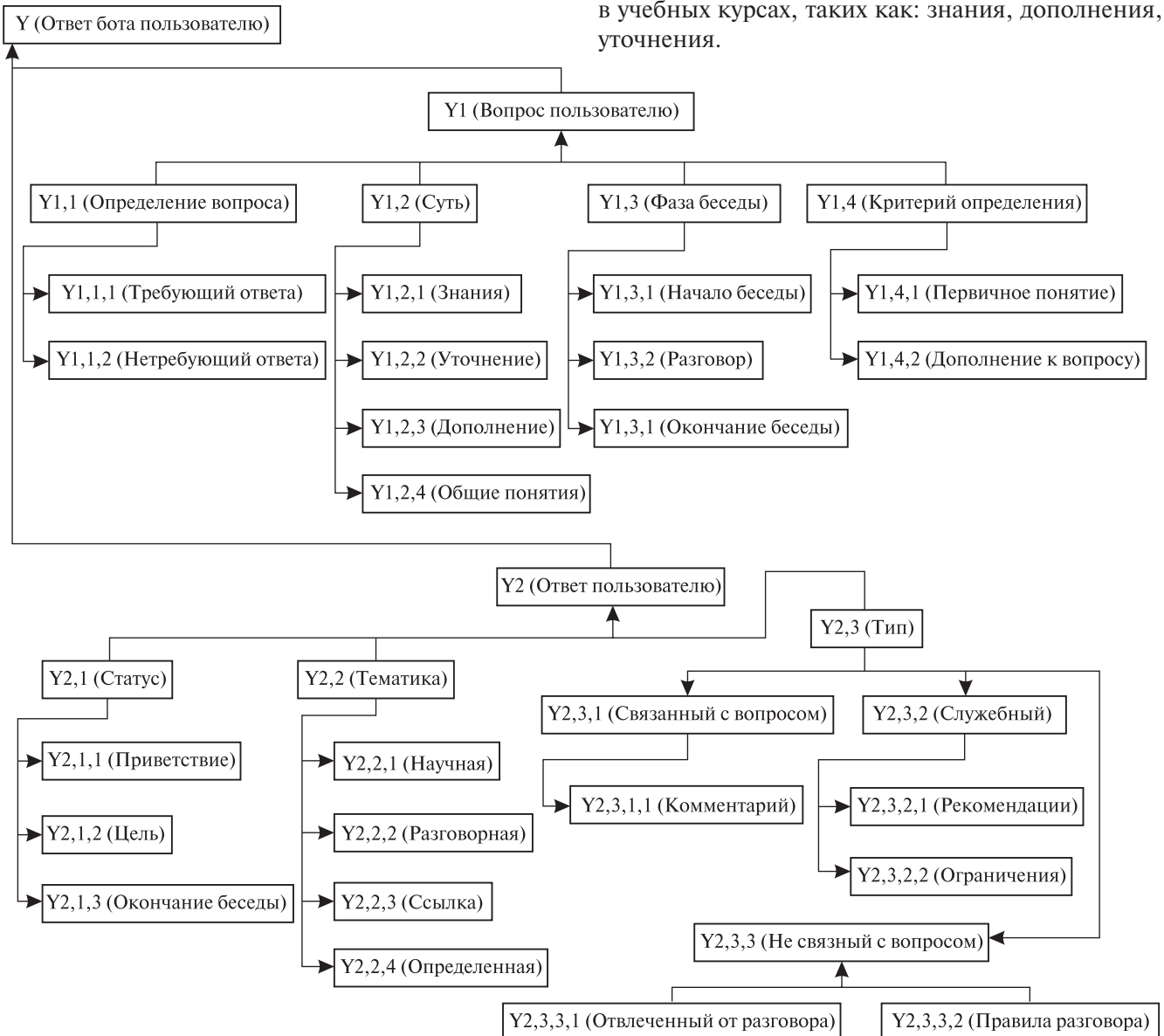


Рис. 2. Онтологическая модель разговора

Предполагается возможность уточнения у бота данного ответа и отнесения вопроса к классу критерия, расклассифицировав его на два подтипа: первичное восприятие вопроса системой или дополнение к ответу $Y_{1,4}$ (критерий определения). Тем самым происходит запись вопроса в класс ответа по уже заранее известной тематике. Предлагается также внедрить в систему плагин-ответ и плагин-вопрос $Y_{1,3}$ (статус) – используемые для определения вопроса, не внесенного в стандартную базу ответов. Расклассифицировав вопрос, чат-бот автоматически по такому же алгоритму начинает генерировать ответ пользователю.

Ответ – наиболее значимая часть работы бота, отвечающая на поставленный вопрос пользователем с минимальным расхождением от реальной сути вопроса.

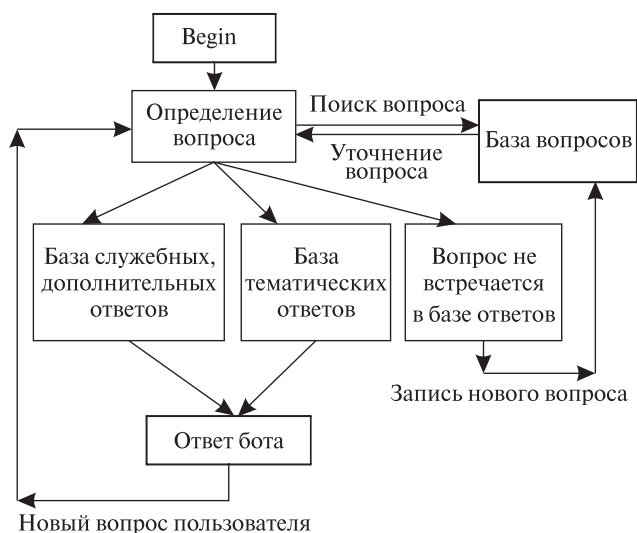


Рис. 3. Схема алгоритма прохода по онтологической модели

Агент, определив множество понятий предметной области X (вопрос) и конечное множество отношений между концептами заданной предметной области R (идентификаторы отношения к ответу) переходит к шагу определения конечного множества функций аксиоматизации, заданных на концептах Φ (ответ) для окончательного представления ответа пользователю $O(1)$. По принципу обработки вопроса агент производит такие же манипуляции с базой имеющихся у него ответов.

С известными параметрами вопроса X, R система относит вопрос к имеющимся ответам. Таксономическая модель ответа предполагает продвижение по ветке Y_2 (ответа) для выявления из базы ответов представления и отношения вопроса к ответу с целью получения Y (ответ бота пользователю). Используются следующие этапы определения наиболее адекватного ответа $Y_{2,1}$ (статус разговора) – как указано выше в разговоре имеются стадии: начало разговора $Y_{2,1,1}$ (определенное действиями агента в системе, для приветствия пользователя);

цель разговора $Y_{2,1,2}$ (реагируя на вопрос или утверждение пользователя, агент придерживается определенных тематик в ответах при разговоре); логическое завершение $Y_{2,1,3}$.

Такая онтологическая модель записывается в XML-файл, который обрабатывает агент и тем самым выполняет разговор с пользователем средствами чата от имени преподавателя.

4. BDI модель программного агента интерактивного взаимодействия пользователей в Jadex

В данной работе используется технология Jadex, которая позволяет моделировать BDI агентов с наборами фактов, целей, планов и другое [5], которые используются для обеспечения поддержки персонализированного обучения.

Функционирование агента интерактивного взаимодействия пользователей осуществляется средствами двух вложенных агентов: модуль коммутации «start_chat» и модуль поддержки пользователей в отсутствии тьютора «chat_bot».

При получении команд вызова чата из приложения СДО вызывается агент «start_chat», который начинает выполнять свои действия согласно установленному приоритету в первоначальных настройках агента в соответствии с планом (рис. 4).

Агент выполняет план «starting», который создает окно чата в соответствии с установленными параметрами и формирует список пользователей, данные о которых получены от агента координатора.

```

<initialstates>
<!-- Используется для установки выполнения первоначальных
    планов агентов -->
<initialstate name=>default<>
    <plans> <initialplan ref=>starting</> </plans>
    <plans> <initialplan ref=>review</> </plans>
</initialstate>
</initialstates>
    
```

Рис. 4. Начальное состояние агента

Список пользователей, определенный в базе знаний системы, представляет собой системный xml-файл, обрабатываемый java-классом «TutorList» (рис. 5). Агент интерактивного взаимодействия в соответствии с полученными адресами передаёт необходимые данные (IP адрес) агенту координатору для установления тестовой связи с тьютором.

```

<beliefs>
    <beliefset name=>sdo_list_tutor class=>TutorList<>
        <fact> new TutorList(<user_tutor.xml>,null) </fact>
    </beliefset>
</beliefs>
    
```

Рис. 5. Знания агента

Файл user_tutor.xml содержит данные, используемые агентами «coordinator» и модулем агента

коммутации «start_chat» для обеспечения интерактивной связи пользователя с тьюторами, проверки статуса и предоставления информации о тьюторе студенту [6].

Определяя статус тьюторов, агент устанавливает графические индикаторы в состояние online/offline, используя план «start_chat», путем отправки сервисного запроса каждому из тьюторов по плану «review» (рис. 8).

```

.....
<goals>
  <!-- Запуск coordinator агента для установления связи -->
  <achievegoalref name=»coordinator»>
    <concrete ref=»startercap.start_coordinator»/>
  </achievegoalref>
  <!-- Запуск Chat_bot агента для выполнения ответа
  пользователю-->
  <achievegoalref name=»start_chat_bot»>
    <concrete ref=»startercap.start_chat_bot»/>
  </achievegoalref>
</goals>
.....
    
```

Рис. 6. Цели агента

Установление связи пользователей модуля агента «start_chat» с терминалами тьюторов осуществляется агентом координатором «coordinator» по запросу пользователя к агенту (рис. 6). Выполнив первоначальные планы, агент переходит в состояние ожидания дальнейших действий пользователя.

По команде пользователя «передать сообщение тьютору» агент пытается передать его, проверяя состояние тьютора. Если тьютор находится в режиме online, агент реагирует на сообщение (рис. 7), последовательно направляя алерт-уведомление на установление связи, используя план «connection» (рис. 8).

```

.....
<events>
  <!-- Обработка запроса на общение с тьютором -->
  <messageevent name=»connect_tutor» direction=»receive»
    type=»fipa»>
    <parameter name=»performative» class=»String»
      direction=»fixed»>
      <value>SFipa.REQUEST</value>
    </parameter>
  </messageevent>
</events>
.....
    
```

Рис. 7. Сообщения, на которые реагирует агент

При подтверждении тьютора на алерт-сообщение, агент устанавливает соединение для общения, используя окно чата.

Если тьютор отказывается принять сообщение или не реагирует, на него находясь в статусе online/offline, агент, используя план «connection», ожидая некоторый промежуток времени, затем выполняет вызов модуля агента «start_chat_bot» для возможного начала диалога с машиной. Модуль «start_chat_bot» работает по заданной онтологической

модели, используя базу знаний, содержащую ответы, вопросы и ключи, определяющие принадлежность ответа к вопросу.

```

.....
<plans>
  <plan name=»starting»>
  <!-- Открытие диалогового окна чата и заполнение контакт
  листа тьюторов -->
    <body>new StartingChat()</body>
  </plan>
  <plan name=»review»>
  <!-- Проверка состояния тьюторов -->
    <body> new CheckSource()</body>
  </plan>
  <plan name=»connection» class=»ConnectionTutor»>
  <!-- Установить связь с выбранным тьютором -->
    <body>new ConnectionTutor()</body>
    <waitqueue> messageevent ref=»connect_tutor»/>
  </waitqueue>
  </plan>
</plans>
.....
    
```

Рис. 8. Планы агента

Выводы

Разработка и внедрение агента интерактивного взаимодействия участников мультиагентной системы дистанционного обучения обеспечивает динамику общения и предоставляет требуемую информацию студенту при отсутствии тьютора. Предложенный в работе программный агент выражен как BDI– модель, для него созданы факты, цели, планы, сообщения, также разработана схема взаимодействия модулей агента и его связь с другими агентами. Реализовано программное приложение с возможностью обработки запросов пользователя: запуск агента, построение GUI, получение информации от агента–координатора для генерирования информации об участниках системы, интерактивная коммутация пользователей, запуск агента чат-бота для оказания поддержки обучаемому, формирования логической связи между полученным запросом и действиями агента.

Модуль агента чат-бот представляет пользователю ответ, заменяя тьютора в его отсутствие, на основании полученного вопроса к системе. Обработка вопроса происходит по разработанной онтологической модели, ограниченной пределами тематики системы ДО.

Таким образом, использование агентных технологий для решения поставленной задачи решает проблему коммутации, используя модуль чата в системе дистанционного образования и взаимодействует с остальными агентами системы, дополняя их функциональные возможности.

Конечным результатом реализации [7,8] агента является подключение к мультиагентной системе дистанционного образования для совместного взаимодействия с другими агентами в соответствии с заданной BDI– моделью.

Список литературы: 1. Copernic Products <http://www.copernic.com>. 2. Mulberry Technologies <http://www.mulberrytech.com>. 3. The World Wide Web Consortium (W3C) <http://www.w3.org>. 4. John Davies, Rudi Studer Semantic Web technologies: trends and research in ontology-based systems, 2006. 5. Alexander Pokahr, Lars Braubach, Andrzej Walczak, Jadex User Guide. 6. Hamburg: Systems Group University of Hamburg, 2005. 7. Программная среда IntelliJ IDEA 6.0 для разработки Java приложений. 8. <http://jadex.sourceforge.net/jadex>.

Поступила в редколлегию 29.09.2008

УДК 519.62

Функціонування програмного агенту інтерактивної взаємодії учасників мультиагентної системи дистанційного навчання / С.А. Марьин, О.П. Мишин, В.Б. Репка // Біоніка інтелекту: наук.-техн. журнал – 2008. – № 2 (69). – С. 95-100.

У статті викладено ідею розробки програмного агента інтерактивної взаємодії користувачів в мультиагентній системі дистанційного навчання. Також представлено схему взаємодії агентів з користувачами, критерій функціонування агента, онтологічну модель розмови агента з користувачами за відсутності тьютора.

Лл. 8. Бібліогр.: 8 найм.

UDK 519.62

Functioning program agent interactive communication of users in distant learning system / S. Marin, A. Mishin, V. Repka // Bionics of Intelligence: Sci. Mag. – 2008. – № 2 (69). – P. 95-100.

In this article the idea of program agent development for interactive communication of users in multi-agents distant learning system is given. Also the scheme of agents' interaction with users, and criteria of agent functioning, and ontological model of conversation between the agent and the tutor (when tutor is absent) are presented.

Fig.8. Ref.: 8 names.