

Федеральное агенство по образованию РФ  
Государственное образовательное учреждение  
высшего профессионального образования  
Уральский государственный университет им. А.М. Горького

Математико-механический факультет  
Кафедра информатики и процессов управления

## **Исследование и разработка интеллектуальной многоагентной обучающей среды**

«Допущен к защите»

\_\_\_\_\_

«    » \_\_\_\_\_ 2010 г.

Квалификационная работа  
на степень магистра наук по направлению  
«математика, компьютерные науки»  
студента группы МГКН-2  
Дернова Григория Сергеевича

\_\_\_\_\_

Научный руководитель:  
к.т.н., доцент кафедры ИПУ  
Авербух Владимир Лазаревич

\_\_\_\_\_

Екатеринбург

2010

## **Реферат**

Дернов Г.С. Исследование и разработка интеллектуальной многоагентной обучающей среды. Квалификационная работа на степень магистра: стр. 67, рис. 4, библиограф. 41 назв.

### **Ключевые слова**

Обучающая среда, электронный учебник, интеллектуальный агент, педагогический агент, многоагентная система, JADE, мультитач-жесты.

### **Описание**

Объектом исследования является технология интеллектуальных агентов. Предметом исследования служат педагогические агенты в составе обучающей системы, обладающие свойством активности, то есть инициируемые изменением среды. Цель исследования заключается в поиске возможностей развития традиционной обучающей среды в направлении многоагентности и интеллектуальности.

Для достижения поставленной цели исследуются существующие технологии проектирования обучающих сред, теоретические вопросы и программный инструментарий разработки агентов и многоагентных систем, разрабатываются функциональные требования к обучающей системе, прототип многоагентного интерфейса, программная модель обучаемого и обучающей системы в целом, исследуется практическая значимость разработанной обучающей среды для решения дидактических задач.

# Оглавление

<b>Введение</b> .....	5
Объект исследования.....	5
Предмет исследования.....	7
Цель и задачи исследования.....	8
Актуальность исследования.....	10
<b>1. Исследование существующих обучающих сред</b> .....	12
1.1. Общий анализ.....	12
1.2. Основные системы.....	13
<b>2. Исследование технологии агентного программирования</b> .....	15
2.1. Алгоритм и структура агента.....	15
2.2. Интеллектуальные агентные системы.....	18
2.3. Многоагентные системы.....	20
2.4. Агентные платформы.....	22
2.5. Агентная платформа JADE.....	25
2.6. Сценарии взаимодействия.....	28
<b>3. Разработка функциональных требований</b> .....	30
<b>4. Разработка программной архитектуры</b> .....	34
4.1. Технические требования.....	34
4.2. Основные компоненты.....	35
4.3. Сценарий взаимодействия.....	36
4.4. Стратегии обучения.....	37
4.5. Реализация понятия.....	39
4.6. Реализация правила.....	40
4.7. Реализация стратегии.....	41

<b>5. Разработка интерфейса</b> .....	43
5.1. Общий анализ.....	43
5.2. Представление агента в интерфейсе.....	44
5.3. Педагогические агенты.....	45
5.4. Прототип интерфейса.....	47
5.5. Динамические характеристики интерфейса.....	50
5.6. Жесты.....	52
5.7. Программная реализация жестов.....	55
<b>Заключение</b> .....	57
<b>Литература</b> .....	61

# **Введение**

## **Объект исследования**

Традиционно под агентом понимается некоторая автономная сущность в составе информационной системы, решающая определенную подзадачу этой системы. В дополнение, на конференции по агентно-ориентированным системам [1], проходившей в 2005 году в Германии, сообществом в качестве основных для агента были утверждены также свойства реактивности (восприятия окружающей среды при помощи датчиков) и дееспособности (наличия исполнительных механизмов, которые способны влиять на среду).

Исторически агенты появились около 20 лет назад и быстро перешли из академических проектов в промышленные разработки [2]. Причина столь активного роста заключается в том, что агенты представляют собой инновационный, не имеющей альтернативы во многих областях, подход к проектированию систем, в котором ведущую роль занимает взаимодействие их друг с другом и с пользователем системы.

С точки зрения исследования вопросов человеко-машинного взаимодействия особый интерес представляют так называемые интеллектуальные агенты.

## **Интеллектуальный агент**

Будем называть интеллектуальным агента, поведение которого «напоминает» человеческое, то есть отвечает известному тесту Тьюринга. Несмотря на недостатки указанного определения, его достаточно, чтобы утверждать, что интеллектуальность является естественным продолжением автономности агентов. Особенно это верно в случае активных агентов, которые обладают инициативой в диалоге с пользователем, демонстрируя

широко известный эффект Пигмалиона (эффект Элизы) [3].

## **Предмет исследования**

В качестве предмета исследования рассматривается особый класс интеллектуальных агентов – активный педагогический агент. По мнению многих экспертов, к настоящему моменту многоагентные системы общего характера исчерпали себя и возникла необходимость в специализированных системах, которые соответствуют конкретным предметным областям и конкретным способам взаимодействия с пользователями.

### **Активный агент как способ взаимодействия**

По способу взаимодействия с пользователем агенты разделяются на пассивных и активных [4]. В первом случае пользователь сам инициирует активность агента, что приводит к выполнению его алгоритма и последующему останова («засыпанию», suspend). Во втором же – активность агента инициируется изменением среды. В данной работе предметом исследования являются именно активные агенты, целесообразность использования которых в интеллектуальных системах продемонстрирована выше.

### **Педагогический агент как предметная область**

Будем понимать под обучающими системами интеллектуальные системы, которые решают дидактические задачи. В случае, если обучающая система проектируется на основе технологии интеллектуальных агентов или использует агентов для человеко-машинного взаимодействия, ее базовые сущности называются педагогическими агентами [5].

Использование агентов оказывается эффективным в случае наличия сложных или динамически изменяющихся структур знаний, что верно отражает ментальную модель обучения.

## **Цель и задачи исследования**

### **Цель исследования**

Главная цель исследования состоит в выявлении основных возможностей развития обучающих систем в русле многоагентности и интеллектуальности, а также изучении сопутствующих инструментов технической реализации этих возможностей.

Следует показать, что наиболее важными в порядке убывания приоритета представляются исследовательские, теоретические и прикладные результаты.

Внимание к исследовательской функции работы обосновывается тем, что следует понять причины бурного роста использования агентов в популярных приложениях в 90-е годы и последующую стогнацию, вплоть до прекращения поддержки инструментария их разработки. К примеру, MS Agent [6] уже не поддерживает многие новые функции браузера.

Внимание к теоретическим положениям обосновывается тем, что, по мнению экспертов [7], область многоагентных систем является значительно менее развитой в теории искусственного интеллекта, чем область, к примеру, представления знаний.

А небольшое внимание к прикладным вопросам взаимодействия агентов можно объяснить тем, что программный инструментарий агентного программирования обладает плохой совместимостью и недостаточной стандартизацией [8], что затрудняет сознательный выбор стратегического средства разработки многоагентных систем.

### **Задачи исследования**

Выделим список вопросов, на которые нужно ответить в ходе исследования, чтобы реализовать поставленную цель:

- 1) В чем заключается специфика и инструментарий технологии агентного программирования?
- 2) Каковы недостатки существующих интеллектуальных обучающих систем?
- 3) Каким требованиям должна удовлетворять разрабатываемая обучающая система?
- 4) Что должна представлять собой программная архитектура обучающей системы?
- 5) Каким должен быть интерфейс взаимодействия с пользователем обучающей системы?

## **Актуальность исследования**

Прежде чем решать задачу, следует понять, является ли она проблемной, то есть не имеющей пока должного решения. В принципе, уже то, что вопрос взаимодействия агентов в составе интеллектуальной системы (в отличие, например, от методов представления знаний) еще не имеет устоявшейся теории и программной поддержки [8] (можно назвать около 20 одинаково популярных платформ для реализации МАС) говорит о проблемности вопроса. Однако, рассмотрим подробнее темы взаимодействия, обучения, и многоагентности в интеллектуальной среде.

## **Проблема взаимодействия**

Актуальность вопроса человеко-машинного взаимодействия подтверждается множеством положений. К примеру, на международной конференции по общим вопросам развития техники в 2004 году учеными Стэнфордского университета была озвучена тенденция смещения интересов в область человеко-машинных систем; Билл Гейтс в качестве лидера компании «Майкрософт» (Microsoft) назвал интерфейс персональных компьютеров главным вызовом для отрасли в 21 веке [9].

Речь идет не только об увеличении роли человека в отношении «человек-компьютер», но и тем, что усложняющийся функционал современных программ уже не может быть эффективно представлен традиционным статичным интерфейсом.

## **Проблема обучения**

Относительно обучения можно отметить возрастающую роль систем дистанционного образования и так называемых информационных технологий обучения (например, мультимедиа-презентаций) [10]. По

сведениям ЮНЕСКО, 70% знаний приобретаются сейчас вне классических форм обучения [11].

Кроме того, в докладах «Института ЮНЕСКО по вопросам ИТ в образовании» наиболее высокие требования предъявляются к инициативности и социализации ученика в образовательном процессе, что может обеспечить только активный процесс взаимодействия с учителем (пусть виртуальным), предоставляемый обучающими системами.

### **Проблема агентов**

Относительно агентов следует упомянуть их бурное внедрение в промышленные системы второй половины 90-х годов и последующий отказ от использования [2]. На наш взгляд, причина заключается в недостаточной специализации поведения агента с точки зрения предметной области, нужд и возможностей пользователя.

### **Вывод**

Казалось бы, комбинация указанных проблем лишь усиливает сложность задачи и требует детальных и дифференцированных исследований вопросов интеллекта, интерфейса и технологий обучения.

На самом деле, они взаимодополняют друг друга. В частности, многоагентность позволяет одновременно моделировать социальное поведение ученика/учителя и является технической возможностью реализации интерактивного интерфейса.

# 1. Исследование существующих обучающих сред

## 1.1. Общий анализ

«Педагогические системы последних поколений интеллектуальны, но не являются интеллектуальными педагогическими системами» [12]. Указанная цитата соответствует центральной идее работы о необходимости введения педагогических понятий и педагогических критериев полезности в алгоритмы интеллектуальных систем.

Среди недостатков были также названы:

- отсутствие организующих систему моделей стратегии и тактики обучения;
- некумулятивность агентов, то есть использование простых агентов (if-then), не учитывающих «историю восприятия».

Последующий анализ обучающих систем, проведенный в ходе данного исследования, дополнительно выявил следующие недостатки:

- обучающие среды используют в основном психологические (ментальные), а не педагогические (дидактические) критерии;
- персонализация присутствует в большинстве обучающих сред, однако информация должна накапливаться системой методом обучения (при помощи адаптации);
- обучающие среды основаны на функциональных, а не интеллектуальных интерфейсах;
- системы отличает слабая мультиагентность, не отражающая теоретический строй и междисциплинарность, что, скорее всего, относится к недостаткам модели знаний системы.

## 1.2. Основные системы

Можно выделить порядка 10 основных интеллектуальных обучающих систем, в авангарде которых находится система IDEAL. В каждой из них заявлены интерактивные и интеллектуальные свойства, однако, следует понимать, что эти термины употребляются иначе.

Отметим ключевые особенности изученных систем:

- NBLE [13] принимает решения на основе графа, в вершинах которого расположены оценки за тему;
- EON [14] использует аналогичную систему - граф допущенных ошибок;
- FLUTE [15] строит сеть на основе процентов прогресса;
- KBS [16] является, прежде всего, адаптивной к целям обучающегося;
- IDEAL [17] – адаптация материала происходит по уровням, используя сети Байеса (метод принятия решений);

Таким образом, можно сделать вывод, что абсолютное большинство обучающих систем использует оверлейные (векторные и сетевые) некумулятивные модели обучаемого, а также учитывает уровень знаний и структуру курса, меньшее число анализирует психотип и качество выполнения заданий, и, наконец, практически никакие системы не ставят в основу стиль и метод обучения.

### Система IDEAL

Систему IDEAL [17] (Intelligent distributed environment for active learning) можно считать одной из наиболее передовых среди обучающих систем.

Наиболее важным достоинством системы является открытая архитектура, доступная для редактирования преподавателем курса.

Кроме того, систему отличает:

- персонализация студента по профилю уровня овладения материалом;
- многоагентность;
- планирование расписания обучения: агентом – по степени готовности студента или студентом – вручную при помощи диаграммы тем для повторения и изучения;
- возможность сборки лекции покомпонентно;

Система построена на базе педагогических агентов. Однако, на наш взгляд, выбор для решения этой задачи языка JavaSpace следует признать не совсем удачным.

В системе существует несколько агентов:

- персональный агент студента, который использует сеть Байеса показателей продуктивности – осуществляет распределение по уровням знаний по изученным к настоящему моменту главам;
- агенты курсов и учебных блоков со знаниями технологий курса;
- учебный агент, который синтезирует двух предыдущих, понимает и генерирует язык;
- групповой агент, который создает групповой профиль.

## **2. Технология агентного программирования**

### **2.1. Алгоритм и структура агента**

#### **Общий анализ**

Для решения задачи программирования агентов можно использовать традиционную объектно-ориентированную парадигму. Однако, в случае экспертных систем, продуктивнее сначала определить знания в декларативном виде в качестве сущностей и отношений, а потом обрабатывать их машиной вывода, подбирающей последовательность действий на основе исходных условий. Учитывая, что количество алгоритмов гораздо меньше количества постановок задач, можно значительно упростить разработку за счет разделения описания задачи от поиска ее решения.

Рассмотрим особенности каждой парадигмы в отдельности [18].

#### **Объектно-ориентированная парадигма**

- логика
  - восприятие
  - действие
  - обучение
- взаимодействие
  - коммуникация
  - кооперация или конкуренция (в зависимости от общности цели)
  - управление другими агентами
- целенаправленность (включение агента в систему для решения общей задачи)

#### **Экспертная парадигма**

- представление знаний, модельное и визуальное
- инженерия знаний
- методы принятия решений

### **Общая структура агента**

Помимо алгоритма и функции полезности, структурообразующими для агента, находящегося в среде, являются датчики для ее восприятия и исполнительные механизмы для действия в среду [19]. В случае примера с таксистом средой служат дороги и другие автомобили, исполнительными механизмами - руль и педали, а датчиками - спидометр.

### **Реактивность и адаптация**

Под реактивностью агента понимается способность отвечать на действия среды [20]. Общая схема взаимодействия агента со средой выглядит так: `when Event if Condition then Action`. Для того, чтобы классифицировать виды оказываемых на агента внешних воздействий, агент должен моделировать окружающую среду. В случае полностью наблюдаемой среды агент называется «простым» и логика его реакции сводится к `if Condition then Action`.

### **Планирование**

Планирование – это способность синтезировать агентом цепочки действий, необходимых для достижения целевого состояния.

### **Обучение (машинное)**

Обучением – это процесс, в результате которого формируются необходимые цепочки ответных действий агента на историю воздействия. Таким образом обеспечивается адаптивность системы – настройка параметров и структуры системы с целью достижения требуемого

качества. Основным механизмом обучения служит генератор проблем.

## **2.2. Интеллектуальные агентные системы**

### **Общий анализ**

Традиционно под искусственным интеллектом (artificial intelligence) понимается алгоритм, решающий некоторую мыслительную задачу. Это бихевиористская концепция, или концепция «черного ящика», которая избегает исследования самих мыслительных процессов. Наряду с определением интеллектуальности при помощи теста Тьюринга [21], для агентного взаимодействия она оставляет больше вопросов, чем ответов.

### **«Возникающий» интеллект**

Ключевым моментом в проектировании многоагентных систем является скачок от дифференциальных характеристик, представленных индивидуальным поведением агента, к интегративным, то есть решению интеллектуальной задачи. Соответствующий переход в системном анализе получил «возникающий» интеллект (emergent intelligence) [22]. К примеру, каким образом на основе определения простого поведения муравья в известной системе StarLogo удастся достичь их объединения в коллектив?

Столь важен этот факт еще и потому, что не только агенты влияют на возникающие структуры – верно и обратное. К примеру, аплодисменты в зале начинаются неорганизованно и децентрализованно, с добавлением новых участников процесс стабилизируется к определенному ритму и с этого момента новые участники лишь подчиняются сложившемуся ритму.

Само понятие скачка означает, что сложившиеся социальные отношения, относящиеся к системному измерению (то есть многоагентной системе), основываются на межагентном взаимодействии, но не являются их прямой суммой.

## **Оптимизационная модель**

Для разрешения этой проблемы предлагается использовать так называемую оптимизационную модель искусственного интеллекта. В этом случае агент оперирует понятиями (реализуемыми как переменные), закодированными человеком, и в зависимости от истории («последовательности актов восприятий») и текущего состояния, которое само интерпретируется понятийно, стремится к наиболее эффективному «действию в среде»). При этом критерий эффективности («функция полезности») задается человеком.

Соответствующая модель поддерживается большим математическим аппаратом: теорией оптимального управления, теорией принятия решений, теорией игр. Однако, в данном исследовании центральным является не синтаксический, а семантический и прагматический аспекты системы.

Таким образом, главным критерием ИИ служит вариативность его действий (выбор на множестве). К примеру, автомобиль должен добраться в определенное место за минимальное время, варьируя выбор маршрута и соблюдая ПДД. В отличие от традиционных программ, в случае разработки интеллектуального алгоритма ведущее значение приобретает не само решение, а правильная постановка задачи и поиск оптимального решения.

## **2.3. Многоагентные системы**

### **Определение**

Многоагентной называется система, решающая одну задачу несколькими агентами методом передачи знаний и задач (действий) между ними [23]. Ее агенты являются ограниченными, то есть не знают о всей системе, и децентрализованными, то есть не управляют всей системой. В случае разнородности входящих в нее агентов по структуре или поведению многоагентная система называется гетерогенной, иначе – гомогенной.

### **Объекты программирования**

- примитивы агента
- цели (состояние агента во времени, влияние выполнения задачи на состояние, способы достижения цели и что происходит, если их нельзя достигнуть)
- организация
- окружение
- взаимодействие

### **Методы программирования**

В силу децентрализации с точки зрения разработчика многоагентные интеллектуальные системы отличаются необходимостью исследования процессов не «сверху вниз», а «снизу вверх» [24]. Этот способ сложнее, но гораздо устойчивее.

С одной стороны, некоторая комбинация агентов, работающих в составе системы, уже образует некоторую общую функцию полезности  $F1$ . С другой стороны, к системе со стороны предъявляются функциональные требования, то есть функция полезности  $F2$ . Поэтому перед разработчиком

многоагентной системы стоит следующий фундаментальный вопрос: как скорректировать индивидуальное поведение агентов так, чтобы максимизация F1 как можно больше коррелировало с максимизацией F2. Поэтому в разработке МАС большое внимание уделяется средствам анализа поведения системы.

## **2.4. Агентные платформы**

### **Определение**

Агентная платформа - базовый инструмент разработки многоагентных систем, позволяющий создавать и уничтожать, интерпретировать, запускать и перемещать агентов [25].

Таким образом, рассмотрим основные функции агентных платформ:

- взаимодействие агентов;
- передача сообщений между агентами внутри платформы (на различных уровнях: на уровне сетевых пакетов, сообщений какого-либо языка общения, протоколов общения) и между разными платформами;
- поддержка онтологий;
- управление агентами, их жизненными циклами;
- поиск агентов и данных о них внутри системы;
- безопасность.

### **Недостатки современных платформ**

Можно утверждать, что на сегодняшний день не существует полноценной агентной платформы, соответствующей требованиям построения агентов [26]. С точки зрения принципов распределенного объектно-ориентированного программирования необходимость передачи методов может быть существенно сокращена в том случае, если может быть обеспечен удаленный доступ к общим методам посредством передачи ссылок на удаленные объекты, данных экземпляров этих объектов и их состояний.

Однако, в дополнение к концепции ООП, каждый агент имеет возможность создания копий самого себя с полной или ограниченной

функциональностью, обеспечивая возможность настройки на среду путем исключения неэффективных методов и замены их новыми.

Таким образом, традиционная для объектно-ориентированного программирования схема класс-объект нарушается, так как агент имеет возможность постоянного изменения сценария поведения без его изменения в родительском классе. Многозначное наследование позволяет создавать экземпляры агентов, смешивая сценарии поведения, схемы наследования и атрибуты, определенные в родительских классах.

Следовательно, система разработки, которая бы полностью соответствовала принципам построения агентов, должна соответствовать таким требованиям [27]:

- обеспечение перенесения кода на различные платформы;
- доступность на многих платформах;
- поддержка сетевого взаимодействия;
- многопоточковая обработка.

### **Основные представители платформ**

Сравнение агентных платформ будем проводить по следующим показателям [28, 29]:

- области применения;
- технологии (языку, стандартам организации FIPA);
- сообществу разработчиков;
- расширяемости (API и плагины);
- интеграции с корпоративными системами;
- документации;
- лицензии;
- связи с коммерческими структурами и примерами проектов.

Рассмотрим наиболее популярные агентные платформы для разработки и их недостатки:

- JADE
- Coguaar (самый крупный проект, поддерживается DARPA)
- Aglobe (недостаток – недостаточная поддержка FIPA-рекомендаций)
- Jason (недостаток – использует собственный язык AgentSpeak для описания агентов)
- Jack (недостаток – требует коммерческая лицензии для использования)

Кроме того, важную роль для выявления закономерностей поведения многоагентной системы играют агентные платформы, предназначенные для моделирования: MASON, RePast, Ascape, NetLogo.

## 2.5. Агентная платформа JADE

### Общий анализ

JADE (Java Agent DEvelopment Framework) на данный момент является наиболее популярной агентной платформой [30]. Проанализируем ее возможности по следующему плану:

- область применения - неограничена: веб, мобильные устройства, промышленное планирование и логистика, научные исследования, экспертные системы;
- технологии: Java SE/ME, поддержка FIPA;
- интеграция: Java EE, CORBA, XML;
- документация: хорошая;
- лицензия: LGPL;
- потребители: Motorola, Prologtor GmbH, France Telecom, Telecom Italia;
- сообщество разработчиков: зрелое, потому что долгое время JADE была единственной платформой после гибели платформ первых поколений;
- инструментарий разработки: реализация, тестирование и отладка, развертывание (отладка обработчиков поведения, сниффер сообщений, средства тестирования, управление платформой и агентами), существует реализация для IDE Eclipse (основного инструмента разработки на Java);
- реализация: язык описания Java, механизмы интеллекта доступны через сторонние плагины BDI и FSM, есть библиотека протоколов согласно

стандартам FIPA, поддержка онтологий, плагины Semantic Web и Web Services;

### **Примитивы пакета взаимодействия**

Большой интерес с точки зрения технологии агентного программирования представляет стандарт FIPA о взаимодействии агентов. Аббревиатура ACL обозначает язык взаимодействия агентов.

Его примитивами являются слова `inform` и `request`, состоящие из предусловия (являются необходимо истинными для достижения цели) и цели отправителя («рационального эффекта»).

Предусловие `inform` (утверждения) в том, что отправитель:

- считает его истинным;
- намерен, чтобы получатель поверил в его истинность;
- не считает, что получатель осведомлен об его истинности.

Предусловие `request` (действия) в том, что отправитель:

- намерен, чтобы было выполнено действие, указанное в содержании;
- считает, что получатель способен выполнить это действие;
- не считает, что получатель уже намерен выполнить действие.

Само сообщение выглядит следующим образом:

( <тип сообщения>

```
:sender           // отправитель
:receiver         // получатель(и)
:content          // содержание
:reply-with       // метка исходящего сообщения
:in-reply-to      // ссылка на входящее сообщение
:replyBy          // лимит времени на ответ
:language         // язык
```

```
:ontology          // онтология
:protocol          // протокол общения
:conversation-id   // идентификатор разговора
)
```

## 2.6. Сценарии взаимодействия

Рассмотрим примитив межагентного общения: отправитель сообщения  $S$  желает проинформировать получателя  $H$  об истинности высказывания  $P$  с использованием слов  $W$  [31].

И представим его дискретный сценарий:

- 1) намерение (отправитель  $S$  решает, что высказывание  $P$  должно быть передано получателю  $H$ )
- 2) выработка (преобразование высказывания  $P$  во фрагмент речи, обеспечивающий высокую вероятность восстановления смысла высказывания)
- 3) синтез (физическая реализация слов  $W_1$ , то есть  $W_1'$ )
- 4) восприятие (получатель  $H$  воспринимает  $W_1'$  как  $W_2'$  и расшифровывает как  $W_2$ )
- 5) анализ (слова  $W_2$  имеют толкования  $P_1, \dots, P_n$ )
  - a) синтаксический (построение дерева синтаксического вывода, в котором внутренние узлы являются словосочетаниями, а внутренние - словами)
  - b) семантический (извлечения смысла фразы в некотором языке представления)
  - c) прагматический (зависимость значения слова от контекста, то есть способа употребления)
- 6) устранение неоднозначности (получение толкования  $P_i$ ):

- a) модель мира (вероятность истинности в мире)
  - b) мыслительная модель (вероятность передачи истинного высказывания агентом при условии, что факт имел место)
  - c) языковая модель (вероятность выбора определенного набора слов при наличии намерения сообщить высказывание)
- 7) усвоение ( $P_i$  становится для агента более истинным или более ложным)

### **3. Разработка функциональных требований**

#### **Общий анализ**

В результате анализа педагогической литературы и консультаций с преподавателями, использующими обучающие системы в своей деятельности, были выявлены основные дидактические требования к подобным системам.

Однако, не только педагогика формулирует принципы для информационных наук, верно и обратное – информатика порождает новую педагогику. Таким образом, следует сначала определиться с техническими возможностями и требованиями к системе. В связи с этим фактом последнее время получила развитие профессия «тьютора» – человека, подбирающего технологии исходя из педагогических задач.

#### **Технические возможности**

Главным техническим требованием к системе в ходе данного исследования является интеллектуальность агента. Данное выше определение позволяет называть педагогического агента интеллектуальным, если он оперирует дидактическими параметрами и максимизирует дидактический критерий полезности.

Во-вторых, в связи с развитием дистанционного образования и разумностью требования совместимости платформ, предпочтение следует отдать веб-технологиям.

В-третьих, технологии позволяют по-новому демонстрировать теории: индукцией (от частного к общему), дедукцией (от общего к частному), абдукцией (понимание частного на основе общего).

Теперь обратимся непосредственно к требованиям современной педагогики по отношению к основным ролям: учителя («как учить»), ученика, автора («что учить»).

### **Активное взаимодействие**

*Инициативность в процессе обучения.* Обучающая система должна способствовать повышению роли студента в учебном процессе, способствовать проявлению ответственности за процесс и результаты обучения.

*Активная работа с материалами.* Студент должен избавиться от роли пассивного слушателя, активно работать с текстом. В частности, выделять его, оставлять заметки на полях.

*Реализация проблемного метода обучения (Дьюи).* Ученик ставится в проблемную ситуацию, формулирует и преодолевает ее, используя теорию в качестве инструмента.

### **Многоуровневость**

*По степени образованности.* Учебник должен быть одновременно нескучным троечнику и интересным отличнику, предоставляя возможность углубить свои знания в интересующей области. Например, классический труд «Теория справедливости» Роулза изучается и как учебник, и как монография; математический анализ одни используют на уровне понятий и теорем, другие – обязательно добавляют изучение доказательств.

*По задачам обучения.* Один и тот же ученик может использовать учебник для углубления знаний или подготовки к экзамену, поэтому необходим разный уровень изложения. Например, «Эволюция физики» Эйнштейна и Инфельда излагается без формул, поэтому используется гуманитариями

как доступный учебник, а физиками – для иллюстрации интересных положений.

*Дидактическая.* Сформулированный Каменским принцип от простого к сложному позволяет постепенно раскрыть содержание глав и обратно — синтезировать их содержание (выводы). Современная дидактика настаивает на уменьшении объема учебника, увеличении его концептуальной насыщенности и выводе все большего материала за рамки обязательного с присутствием в дополнениях. Дополнениями являются именной и предметный указатели, пособия (хрестоматии, задачник, тетрадь) и пр.

### **Развитая концептуальность**

Точное представление научно-теоретического строя предмета считается главным содержанием учебника: ведь наука плюралистична, но ее теории неравнозначны. В учебнике поле постоянного напряжения – это теория, о которой идет речь: ученик должен осознавать ее основные принципы (фрейм понимания) и/или понятия. Существующие технологии позволяют реализовать разнообразную связность теорий.

Кроме того, внутри науки теории ранжируются по области применения. В экономике если говорят о государстве, на первый план выходит кейнсианство, об анализе цен – неоклассика. Для ранжирования можно использовать метод Саати.

Ранжируются также внутри предметной области – по концептуальной зрелости: более развитые теории позволяют интерпретировать менее развитые (Поппер).

Особое внимание следует уделить вопросу содержания исторического курса учебника. С точки зрения современной педагогики не рекомендуется включать его до теоретического, потому что так исторические события

подаются как очевидные, а они должны пониматься с позиции современных теорий. С другой стороны, после теоретического курса он не так интересен. Опять же, существующие технологии позволяют вплетать исторический материал в качестве демонстрации роста знания в теоретический курс и при этом иметь возможность вычленить этот материал (к примеру, по имени исследователя).

### **Междисциплинарность**

Современное образование междисциплинарно: например, экономика бедна без математических формул, полет стрелы из механики прекрасно иллюстрирует производную. В этом смысле учебник представляет собой энциклопедию, что плохо согласуется с объемом традиционного бумажного учебника.

В энциклопедии особое внимание уделяется языку, употреблению слов, их значению в составе разных теорий. Энциклопедичность делает знание более действенным, усиливает эффект основной теории, указывает место в системе наук, но угрожает превратить текст в «кашу».

### **Наличие обратной связи**

Преподаватель должен обладать как можно более полной информацией о возникших затруднениях студента в ходе обучения и о месте их возникновения.

## 4. Разработка программной архитектуры

### 4.1. Технические требования

Активный учебник построен на базе традиционной клиент-серверной архитектуры.

Рассмотрим основные программные средства, лежащие в ее основе:

- 1) на стороне сервера используются сервлеты (Java Servlet, JSP). Выбор сервлетов (вместо альтернативных ASP и PHP) оказывается удобен с точки зрения их совместного использования с библиотекой JADE. В качестве сервера тестировался Tomcat с использованием базы данных MySQL.
- 2) На стороне клиента применяется JavaScript. Прежде всего, это вызвано необходимостью использования AJAX (соответственно, библиотеки jquery) для загрузки компонентов документа без перезагрузки страницы целиком.
- 3) Для разметки документа используется стандарт XHTML.

## **4.2. Основные компоненты**

### **Модель знаний**

В основе активного учебника лежит некоторая модель знаний, представляющая собой семантическую сеть понятий. Причем каждое понятие либо связано с некоторым множеством html-страниц (являющихся, по сути текстом аннотации и текстом подробного описания), либо является абстрактным (не связанным с конкретными текстами).

### **Модель студента**

Следующим базовым компонентом учебника является модель студента, которая представляет собой некоторое множество концептов, связанных отношениями, которые отражают различные характеристики пользователя.

Значительной частью модели студента являются понятия модели знаний. Их значения интерпретируются как уровень знаний того или иного понятия и лежат в пределах от 0 до 100.

### **Контент**

Любой блок контента (абзац, глава, раздел и пр.) может быть помечен множеством понятий (ключевых слов). В случае, когда активируется событие на стороне клиента, связанное с указанным блоком (открытие, листание, положение курсора и пр.), происходит активация действий, связанных с указанными понятиями. К примеру, это может быть увеличение уровня знаний этого концепта.

Действия задаются преподавателем и могут иметь рекурсивный характер (то есть приводить к активации действий по всему дереву понятий). К примеру, увеличение знаний по главе может приводить к увеличению знаний по конкретному абзацу главы.

### 4.3. Сценарии взаимодействия

Учебник содержит как статическую информацию, которая загружается всегда, так и динамическую, представление которой зависит от заданных преподавателем стратегий обучения.

Рассмотрим типичный сценарий взаимодействия пользователя с системой:

- 1) Пользователь совершает некоторое действие на стороне клиента (листание, положение курсора, открытие страницы и пр.), которое приводит к активации определенного события.
- 2) Система фиксирует блок контента, который связан с событием (абзац, глава). К примеру, положение курсора на уровне абзаца может приводить к активации события, связанного с этим абзацем.
- 3) Указанный блок маркирован некоторым множеством понятий (ключевых слов). Таким образом, пользователь фактически запрашивает некоторое множество понятий.
- 4) Каждое понятие имеет несколько связанных с ним страниц. В зависимости от текущей модели пользователя и его текущего состояния, множество связанных с понятием страниц сортируется с точки зрения того, насколько предпочтительной является соответствующая страницы с точки зрения системы. Соответствующее значение вычисляется на основе правил, определенных в стратегиях обучения.
- 5) С каждым понятием связаны правила в виде событие-условие-действие, которые активируются и в результате обновляют модель пользователя.

## **4.4. Стратегии обучения**

Одним из преимуществ активного учебника является то, что стратегии обучения не являются predetermined и задаются преподавателем. В текущей версии системы реализованы следующие стратегии обучения.

### **Способ представления информации**

Каждое понятие может иметь несколько видов представления: текстовый и визуальный, графический (график) и формульный и пр. В зависимости от текущего состояния модели студента выбирается один из видов представления.

### **Тип информации по отношению к понятию**

Дополнительная информация, которая показывается в области агента, может иметь различный тип по отношению к проиндексированному понятию.

В частности, предопределим следующие виды информации:

- вид «более подробная» (которая имеет дополнительное разбиение на уровни «подробности»)
- вид «пример» (иллюстрирующий содержание понятия)
- вид «проблема» (особый вид текста, содержащий ссылки на связанные с ним тексты – то есть текст, требующий дополнительного изучения материала)
- вид «задача» и пр.

### **Уровень активности**

Данная стратегия оценивает предпочтительный способ, которым студент решает практическую задачу. В одном случае, студент предпочитает сначала изучить всю доступную информацию по теме и лишь потом переходить к решению задачи. Обратная ситуация – когда студент сначала пытается собственно решить задачу и лишь в случае неудачи переходит к поиску новой информации.

### **Характер усвоения материала**

Можно выделить как минимум две стратегии усвоения – интенсивную и экстенсивную. В первом случае студент предпочитает подробно изучить тему, прежде чем переходить к изучению других тем. Во втором случае, предпочитает сначала увидеть общую картинку содержания и лишь потом переходить к конкретизации материала.

## 4.5. Реализация понятия

Рассмотрим реализацию базового объекта активного учебника – понятия.

Понятие представляет собой запись в XML-файле, состоящую из следующих строк:

- `<id>`. Тип `Integer`. Идентификатор понятия в базе данных.
- `<name>`. Тип `String`. Название понятия.
- `<alt>`. Тип `String`. Описание понятия (используется в системе поиска, при выводе ключевых слов, к качестве alt-описания для слов и изображений).
- `<text>`. Тип `Array[String]`. Множество ассоциированных с понятием текстов агента.

Кроме того, любое понятие обладает несколькими атрибутами:

- `name`. Тип `String`. Имя атрибута (`knowledge`, `count`...).
- `type`. Тип `String`. Тип атрибута (`boolean`, `integer`, `real`, `string`).
- `alt`. Тип `String`. Описание атрибута.
- `default`. Значение по умолчанию (может быть также выражением).
- `temp`. Тип `Boolean`. Параметр фиксирует, запоминается ли значение в модели пользователя или является временным.

## 4.6. Реализация правила

Правила состоят из трех компонент:

- Событие (event)
- Условие (condition)
- Действие (action)

Таким образом, правила задаются в формате `when Event if Condition then Action else Action`.

Соответственно, используются теги:

- `<if>`. Тип `Boolean`. Логическое выражение, которое вычисляется при возникновении события.
- `<then>`. Набор действий `<action>`, который выполняется в случае, если условие является верным.
- `<else>`. Набор действий `<action>`, который выполняется в случае, если условие является ложным.

Кроме того, правила могут иметь атрибут `loop`, которые принимает значение типа `Boolean` и означает необходимость распространения правила на связанные атрибуты. Важно понимать, что установка этого параметра в значение `true` может приводить к появлению потенциально бесконечных циклов. По умолчанию атрибут имеет значение `“false”`.

## 4.7. Реализация стратегии

Стратегии обучения задаются в формате XML.

В качестве примера рассмотрим схему, соответствующую понятию «креативность». Определим правила, которые назначают несколько видов обновления модели пользователя в зависимости от текущего уровня освоения понятия студентом.

```
<term>
  <id>221</id>
  <name>Creativity</name>
  <alt>Something about creativity</alt>
  <text>creativity.htm</text>
  <if>this.level > 80</if>
  <param name="scroll" type="bool" temp="false">
    <algo loop="true">
      <if>this.level < 100</if>
      <then>
        <make>
          <conceptID>285</conceptID>
          <param>level</param>
          <do>10</do>
        </make>
      </then>
      <else>
        <make>
          <termID>122</termID>
          <param>level</param>
          <do>10</do>
        </make>
        <make>
          <termID>31</termID>
          <param>knowledge</param>
          <do>10</do>
        </make>
      </else>
    </algo>
  </param>
</term>
```

</param>  
</term>

## 5. Разработка интерфейса

### 5.1. Общий анализ

Требование многоагентности, которое предъявляется нами по отношению к обучающей системе, непосредственным образом влияет на специфику интерфейса активного учебника. В частности, большое значение приобретает динамика взаимодействия пользователя с системой, разворачивающаяся во времени.

Поэтому применение традиционного (функционального) интерфейса, который предоставляет инструменты по требованию (клику и пр.) в виде иконок, оказывается малоэффективным. В отличие от него, интеллектуальный (агентный) интерфейс активного учебника должен предоставлять функционал в связи с а) событиями и б) контекстом.

Кроме того, активный учебник представляет собой огромную базу знаний. Поэтому интерактивное представление элементов интерфейса позволяет преодолеть конфликт возрастающих возможностей программ и ограниченных способностей восприятия и памяти человека.

В более широком смысле агентный интерфейс представляет собой переход от семантического визуального текста (в форме иконки и соответствующего ей значения) к прагматическому тексту (в котором функциональность иконки зависит от ее употребления в «визуальном языке»).

## 5.2. Представление агента в интерфейсе

Следует особо подчеркнуть вопрос представления агента в интерфейсе активного учебника. Он требует дальнейшего уточнения авторской позиции. В своих рассуждениях мы будем использовать теорию метафор В.Л. Авербуха [32].

С одной стороны, можно привести ряд аргументов в пользу реализации антропоморфных метафор агента:

- интеллектуальные агенты способны моделировать разумного собеседника (эффект Пигмалиона [3]), реализуя тем самым метафору диалога
- агенты способны эффективно моделировать социальное поведение

Кроме того, многие исследования, посвященные педагогическим агентам (п. 7.3), указывают на распространенность этого подхода.

Однако, в интерфейсе активного учебника мы ушли от принципа антропоморфности по ряду причин [33]. Главной из них является то, что антропоморфность является дополнительной образностью, которая может как способствовать, так и противоречить дидактическим задачам.

Нам представляется, что наиболее рафинированной метафорой агента в интерфейсе может служить метафора ассистента, как ее определил один из создателей теории агентов – Алан Кей [34]: как «программу, которая при получении задания способна поставить себя на место пользователя, а в случае попадания в тупик лишь задает ему уточняющие вопросы».

### 5.3. Педагогические агенты

Будем называть педагогическим агента, реализующего дидактические функции в обучающей системе. Для дальнейшего изложения нам важно выделить ряд существенных характеристик агента [35].

Во-первых, агенты обладают различными способностями:

- общими
  - координацией голоса и действий
  - вводом естественного языка
  - персонификацией
- специализированными
  - слежение за действиями студента
  - реализация методов деятельностной педагогики

Во-вторых, педагогические агенты зачастую моделируют реальное поведение человека. В частности, известные анимированные агенты помогают в навигации по темам, обращают внимание студента на интересные моменты, проводят презентации: Стив (Steve, живет в трехмерном мультимедиа), Адэле (Adele, живет в веб-среде), Герман (Herman the Bug, общительный «всезнайка»).

Крайне интересен опыт S. Marsella [36], использующего для обучения интерактивную педагогическую драму «Яркие идеи Кармен» (Carmen's bright ideas), в которой агенты являются актерами, а ученики - актерами либо зрителями. В ней пользователи участвуют в разрешении проблем по типу квеста. В качестве платформы для реализации выбран Тезпиан (Thespiian).

Позже метафора драмы получила интересное продолжение в системе языкового обучения ТактЛэнг (TactLang), которая изменяет свое поведение в зависимости от педагогических целей, языка, страны... Логика

продукта реализуется при помощи ПсихСим (PsychSim) (многоагентной системы для ментального моделирования).

Отдельно следует выделить класс агентов для обучения на равных (peer learning). Эффект «равнозначности» может достигаться несколькими способами:

- правдоподобием обстановки, в которой присутствует агент и пользователь на экране
- самим агентом, ведущим диалог с пользователем, используя мимику, эмоции...

Для реализации в активном учебнике важен также еще один класс педагогических агентов – демонстрационных. Они используют деятельностную педагогику для изучения принципов, законов, связей. Пользователь, изменяя параметры или программируя агентов, может наблюдать соответствующие изменения в поведении агентов. К примеру, Виленски (Wilensky) реализовал изучение электромагнетизма на платформе НетЛого (NetLogo) – в этом случае агентами выступают элементарные частицы.

На основе приведенного опыта реализации педагогических агентов, перейдем к вопросу разработки интерфейса обучающей среды.

## 5.4. Прототип интерфейса

В вопросе проектирования интерфейса, прежде всего, следует определиться с расположением основных функциональных блоков.

Можно выделить три таких блока (рис. 7.4.1):

- область заголовка
- область контента (основного содержания страницы)
- область агента

The screenshot shows a web browser window with the title 'Активный учебник' (Active textbook) and a search bar with 'Google'. The page content is titled 'The History of Creativity' and features a profile for 'Sir Ken Robinson'. The profile includes a photo of Sir Ken Robinson, a short bio, and a quote: 'challenges the way we're educating our children. He champions a radical rethink of our school systems, to cultivate creativity and acknowledge multiple types'. Below the bio is a section titled 'Bring on the revolution!' with a sub-header 'Мне непонятно' (I don't understand) and a text input field containing the question 'Что такое карта разума?' (What is a mind map?).

Below the text input field is a detailed anatomical diagram of the human brain. The diagram is color-coded and labeled with various parts of the brain. The top view shows the Frontal lobe, Motor speech area of Broca, Temporal lobe, Longitudinal fissure, Premotor area, Precentral gyrus, Postcentral gyrus, Parietal lobe, and Occipital lobe. The side view shows the Parietal lobe, Reading comprehension area, Occipital lobe, Sensory speech area of Wernicke, Cerebellum, Medulla oblongata, and Pons.

Below the diagram is a quote: 'Creative people revel in making something out of nothing, so when a new year and a blank calendar appears, the possibilities seem endless. Suddenly, impossible editors aren't so intimidating, artistic debuts are only one workshop away and sales goals are easy to achieve.'

Below the quote is another paragraph: 'From 1985-89, he was Director of The Arts in Schools Project, a national initiative to develop the arts in primary and secondary schools throughout England and Wales. The project was funded by the National Curriculum'

Рис. 7.4.1 Прототип интерфейса активного учебника

Рассмотрим подробнее содержание каждого блока.

### **Область заголовка**

Данная область содержит:

- заголовок страницы
- навигационные элементы (ссылки):
  - заголовок раздела
  - заголовки предыдущей и следующей страницы в главе

### **Область контента**

Данная область представляет собой основное содержание страницы учебника. Любой абзац текста подсвечивается при наведении, маркируя то, что он может быть выделен пользователем при помощи клика мышки на абзаце. Выделение абзаца запоминается в профиле пользователя.

Необходимо обратить внимание, что некоторые абзацы и картинки помечены слева символом «стрелки», о назначении которой будет сказано выше (п. 7.5).

### **Область агента**

Данная область реализуется на двух уровнях представления: аннотации и подробного описания.

Аннотация представляет собой абзац текста (возможно, с картинкой), имеющий заголовок. При наведении мышкой на область аннотации ее фон подсвечивается, маркируя то, что аннотация является ссылкой.

Необходимо обратить внимание, что некоторые аннотации помечены слева символом «стрелки», о назначении которой будет сказано выше (п. 7.5).

Подробное описание открывается в специализированном вложенном окне (лайтбоксе, lightbox), которое открывается поверх страницы (рис. 7.4.2)

внутри основного окна. Лайтбокс может быть закрыт пользователем при помощи клика на символе «крестик» либо в любом месте окна, не принадлежащем лайтбоксу.

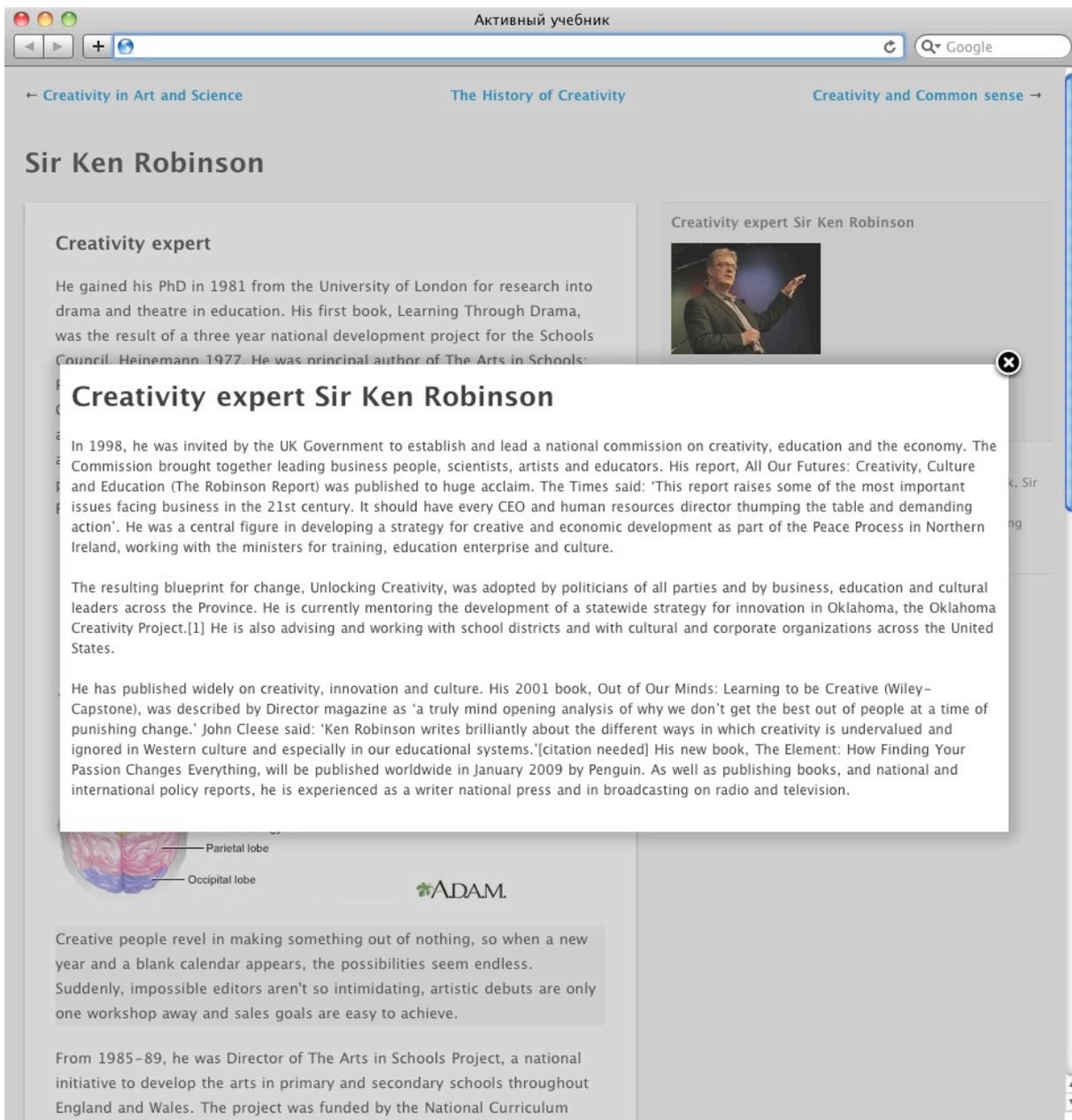


Рис. 7.4.2 Подробное описание открывается в лайтбоксе

## 5.5. Динамические характеристики интерфейса

Кроме статических характеристик интерфейса (расположения его элементов и пр.), существуют динамические характеристики, которые описывают сценарий взаимодействия пользователя с системой (в частности, вопросы навигации пользователя по контенту).

Стандартной технологией для веб-сервисов является навигация на основе гиперссылок.

Она обладает рядом преимуществ:

- не требует дополнительных устройств ввода;
- является сформированной стереотипной привычкой веб-пользователей;
- программно реализована на уровне браузера, не требуя написания дополнительных скриптов

Однако, существуют альтернативные модели навигации, одной из которых является язык жестов. Мы будем рассматривать узкий класс так называемых мультитач-жестов (multitouch) [37]. Они представляют собой жесты, в которых задействовано сразу несколько пальцев одной руки. В качестве устройства ввода таких жестов традиционно используется некоторая (обычно плоская) поверхность. В последнее время появился целый ряд устройств, которые способны обрабатывать такие жесты: разнообразные сенсорные панели, специализированные мыши, устройства с мультитач-экраном.

В контексте активного учебника особый интерес представляют электронные планшеты, которые имеют удобный форм-фактор для чтения электронных книг.

Кроме того, жесты обладают важными для чтения электронных книг преимуществами перед гиперссылочной навигацией:

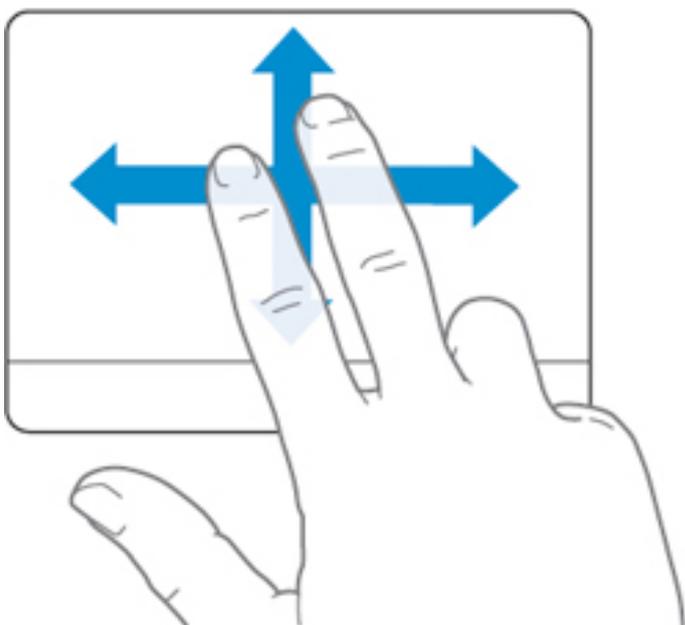
- реализуют ряд стереотипных действий читателя книги, осуществляемых рукой (к примеру, листание страниц);
- жесты позволяют осуществлять навигацию в любом месте экрана, не требуя скроллинга или перемещения по ссылкам (к примеру, для получения оглавления раздела или кратких выводов по тексту);
- жесты требуют от пользователя так называемых «широких действий» (то есть осуществляемых в более широкой области экрана), требуя меньшей концентрации и напряжения, в отличие от «узких действий» (как клик по ссылке)

Указанные аргументы привели нас к следующей концепции интерфейса активного учебника:

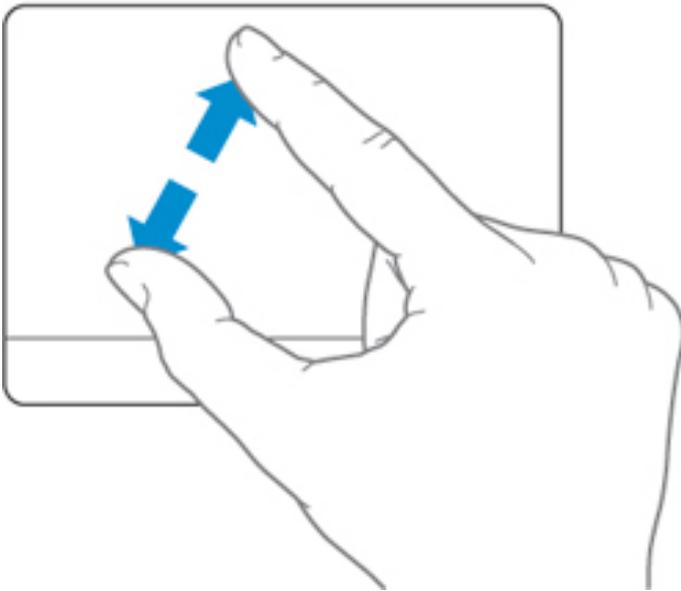
- навигация при помощи мультитач-жестов является предпочтительной, но имеет ряд ограничений;
- гиперссылочная навигация должна быть реализована в качестве базовой, то есть использоваться в случае, если мультитач-жесты являются для пользователя недоступными.

## 5.6. Жесты

Прежде всего, следует определиться с тем, какое множество жестов является наиболее адекватным для использования в активном учебнике. Важно заметить, что многообразие жестов является скорее отрицательной характеристикой интерфейса, чем положительной, потому что жесты требуют запоминания. На наш взгляд, оптимальным является небольшая вариативность реализованных в интерфейсе жестов, которые выполняют различные действия, будучи вызванными в разных контекстах. В активном учебнике мы выделяем две основные группы жестов (см. рис. 7.6.1, 7.6.2).



*Рис. 7.6.1 Двупальцевые жесты «влево», «вправо», «вверх», «вниз»*



*Рис. 7.6.2 Двупальцевые жесты «расширение», «сужение»*

Рассмотрим области действия указанных жестов в интерфейсе активного учебника.

### **Область заголовка**

Двупальцевый жест «вверх» в области заголовка соответствует клику на названии раздела и приводит к появлению лайтбокса с оглавлением раздела.

Двупальцевые жесты «влево» и «вправо» в области заголовка приводят к загрузке соответственно предыдущей и следующей страницы главы.

### **Область контента**

В случае, если абзац или иллюстрация содержит дополнительную информацию (в этом случае они маркированы слева символом «стрелки»), двупальцевый жест «вправо» в области контента приводит к активации соответствующей области агента. Если же абзац или иллюстрация не предусматривает активации, в области в области агента появляется форма

с текстовым (textbox), предназначенная для обратной связи с пользователем.

Двупальцевый жест «расширение» в области абзаца или иллюстрации приводит к появлению особого информационного блока агента, отвечающего за более подробное раскрытие материала (в случае, если такой блок существует).

Двупальцевый жест «сужение» в любом месте области контента приводит к появлению лайтбокса, содержащего краткие выводы по странице.

## 5.7. Программная реализация жестов

Технология жестов реализуется на основе стандартной событийной модели JavaScript DOM [38]. Зарегистрировав мультитач-событие, система постоянно отправляет `TouchEvent` соответствующим элементам DOM.

Мультитач-последовательность начинается, когда палец впервые касается поверхности и заканчивается, когда последний палец снимается с поверхности. Эти события похожи на события, вызываемый мышкой, с одним отличием – на экране в один момент времени имеется несколько прикосновений (`touches`) в нескольких местах экрана.

Важно упомянуть, что браузеры могут самостоятельно «отлавливать» некоторые события. Для того, чтобы стандартное поведение браузера не вмешивалось в код программы, следует применить функцию `event.preventDefault()`.

### События, связанные с жестами

Различают четыре вида событий:

- `touchStart`
- `touchMove`
- `touchEnd`
- `touchCancel`

Прежде всего, событие должно быть зарегистрировано:

```
element.addEventListener("touchstart", touchStart, false);
```

Здесь первый параметр – название обработчика, второй параметр – название соответствующей функции.

Для того, чтобы написать алгоритм обработки жеста, нам необходимо каким-то образом получать все прикосновения, связанные с событием.

Массив «прикосновений» можно получить при помощи `event.touches`.

А положение конкретного «прикосновения» на экране можно получить при помощи параметров `.pageX`, `.pageY` элементов массива.

### **Высокоуровневая обработка жестов**

Эти события использовались в активном учебнике для обработки двупальцевых жестов «влево», «вправо», «вверх», «вниз».

Для реализации же двупальцевых жестов «расширение» и «сужение» более удобным представляется использование более высокоуровневого объекта `GestureEvent`.

Различают три вида соответствующих событий:

- `gestureStart`
- `gestureChange`
- `gestureEnd`

Удобство использования этого объекта состоит в том, что разработчику нет необходимости фиксировать степень «сжатия» – ее можно получить, обратившись к параметру `event.scale`.

## **Заключение**

### **Основные результаты**

Проведенное исследование выявило широкий потенциал использования агентного подхода с точки зрения развития возможностей обучающих сред. Прежде всего, это относится к реализации дидактической стратегии активного обучения и интеллектуального интерфейса взаимодействия пользователя с системой.

В ходе работы были достигнуты следующие результаты:

- 1) Исследована специфика и программный инструментарий технологии агентного программирования.
- 2) Исследованы недостатки существующих интеллектуальных обучающих систем.
- 3) Разработаны функциональные требования к обучающей системе, необходимые для реализации стратегии активного обучения.
- 4) Разработана программная архитектура обучающей системы «Активный учебник».
- 5) Разработан интерфейс взаимодействия с пользователем обучающей системы.

### **Вопросы**

Кроме того, были получены ответы на сформулированные ранее вопросы:

- 1) В чем заключается специфика и инструментарий технологии агентного программирования?

Основное внимание в технологии агентного программирования уделяется вопросу взаимодействия агентов в составе многоагентной системы, которое позволяет решать одну общую задачу методом передачи знаний и задач (действий) между агентами системы.

Существует более 20 одинаково популярных агентных платформ, среди которых мы выделяем JADE. Однако, все из них не лишены недостатков, полученных в качестве наследства от объектно-ориентированного подхода.

2) Каковы недостатки существующих интеллектуальных обучающих систем?

Современные обучающие системы используют недостаточное количество дидактических параметров в своем интеллектуальном поведении и не уделяют должного внимания интерфейсу взаимодействия пользователя с системой.

3) Каким требованиям должна удовлетворять разрабатываемая обучающая система?

Обучающая система должна обеспечивать:

- активный процесс обучения (инициативность, ответственность за результаты обучения)
- адаптивность с точки зрения представления материала учебника(многоуровневость, личностная ориентация)
- развитую концептуальность (представление теоретического строя предмета, междисциплинарность)
- реализацию проблемного метода обучения

4) Что должна представлять собой программная архитектура обучающей системы?

Обучающая система реализована на базе традиционной клиент-серверной архитектуры при помощи кроссплатформенных решений.

На стороне сервера используются сервлеты (JSP), которые получают информацию от агентов, реализованных на агентной платформе JADE. На стороне клиента сообщения агентов динамически загружаются в специальной области HTML-страницы при помощи языка JavaScript с использованием библиотеки jquery (и методов AJAX). Стратегии обучения, модель студента, концептуальная модель и прочие необходимые данные хранятся в XML-файлах.

5) Каким должен быть интерфейс взаимодействия с пользователем обучающей системы?

Интерфейс обучающей системы предоставляет функционал динамически, в связи с событиями и контекстом использования. Показано, что, несмотря на популярность, требование антропоморфности агента является необязательным – в обучающей системе используется более рафинированная метафора ассистента.

Взаимодействие построено на основе мультитач-жестов (на данный момент, двух видов двупальцевых жестов), программно реализуемых при помощи JavaScript DOM. Однако, в случае наличия технических ограничений, возможно использование системы на основе традиционной гиперссылочной навигации.

### **Перспективы работы**

Результаты проведенного исследования планируется использовать в качестве основания для разработки специализированных учебных курсов и

их последующего тестирования с точки зрения эффективности решения дидактических задач.

## Литература

- 1) Torsten Eymann, Franziska Klügl, Winfried Lamersdorf, Matthias Klusch, Michael N. Huhns (Eds.): Multiagent System Technologies, Third German Conference, MATES 2005, Koblenz, Germany, September 11-13, 2005, Proceedings. Lecture Notes in Computer Science 3550 Springer 2005, ISBN 3-540-28740-X
- 2) Russell, S. J. and Norvig, P. 1995 Artificial Intelligence: a Modern Approach. Prentice-Hall, Inc.
- 3) Carreira, J. and Silva, J. G. 1998. Computer Science and the Pygmalion Effect. *Computer* 31, 2 (Feb. 1998), 116-117.
- 4) Jennings, N. R., Sycara, K., and Wooldridge, M. 1998. A Roadmap of Agent Research and Development. *Autonomous Agents and Multi-Agent Systems* 1, 1 (Jan. 1998), 7-38.
- 5) Lester, J. C., Converse, S. A., Kahler, S. E., Barlow, S. T., Stone, B. A., and Bhogal, R. S. 1997. The persona effect: affective impact of animated pedagogical agents. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (Atlanta, Georgia, United States, March 22 - 27, 1997)*. S. Pemberton, Ed. CHI '97. ACM, New York, NY, 359-366.
- 6) Microsoft, C. 1998 *Developing for Microsoft Agent*. Microsoft Press.
- 7) Woolridge, M. and Wooldridge, M. J. 2001 *Introduction to Multiagent Systems*. John Wiley & Sons, Inc.
- 8) Bellifemine, F., Poggi, A., and Rimassa, G. 2001. JADE: a FIPA2000 compliant agent development environment. In *Proceedings of the Fifth international Conference on Autonomous Agents (Montreal, Quebec, Can-*

- ada). AGENTS '01. ACM, New York, NY, 216-217.
- 9) Jorgensen, A. H. and Myers, B. A. 2008. User interface history. In CHI '08 Extended Abstracts on Human Factors in Computing Systems (Florence, Italy, April 05 - 10, 2008). CHI '08. ACM, New York, NY, 2415-2418.
  - 10) Wang, A. J. 2006. IT education in the flattening world. In Proceedings of the 7th Conference on information Technology Education (Minneapolis, Minnesota, USA, October 19 - 21, 2006). SIGITE '06. ACM, New York, NY, 99-104.
  - 11) Mulder, F. and van Weert, T. 2001. IFIP/UNESCO's informatics curriculum framework 2000 for higher education. SIGCSE Bull. 33, 4 (Dec. 2001), 75-83.
  - 12) Dimitrova, V., Mizoguchi, R., Boulay, B. D., Graesser, A., Dimitrova, V., Mizoguchi, R., Boulay, B. D., and Graesser, A. 2009 Artificial Intelligence in Education - Building Learning Systems that Care: from Knowledge Representation to Affective Modelling, Volume 200 Frontiers in Artificial Intelligence and Applications. IOS Press.
  - 13) Nykänen O., Ala-Rantala M., 1997. A Design for Hypermedia-Based Learning Environment.
  - 14) Murray T., 1998. Authoring KnowledgeBased Tutors: Tools for Content, Instructional Strategy, Student Model, and Interface Design. Journal of the Learning Sciences, Vol 7, No 1, 1998, pp. 5-64.
  - 15) Devedzic V., Debenham J., Popovic D, 2000. Teaching Formal Languages by an Intelligent Tutoring System. Educational Technology & Society 3(2) 2000. ISSN 1436-4522
  - 16) Henze N., Nejd W, 1999. Student Modeling in an Active Learning Envi-

ronment using Bayesian Networks.

- 17) Shang Y., Shi H., Chen S., 2001. An Intelligent Distributed Environment for Active Learning.
- 18) Bölöni, L., Luotsinen, L. J., Ekblad, J. N., Fitz-Gibbon, T. R., Houchin, C., Key, J. L., Khan, M., Lyu, J., Nguyen, J., Oleson, R., Stein, G., Vander Weide, S. A., and Trinh, V. 2008. A comparison study of 12 paradigms for developing embodied agents. *Softw. Pract. Exper.* 38, 3 (Mar. 2008), 259-305.
- 19) Hindriks, K. V. and Meyer, J. -. 2009. Toward a programming theory for rational agents. *Autonomous Agents and Multi-Agent Systems* 19, 1 (Aug. 2009), 4-29.
- 20) Van Splunter, S., Wijngaards, N. J., and Brazier, F. M. 2003. Structuring agents for adaptation. In *Adaptive Agents and Multi-Agent Systems: Adaptation and Multi-Agent Learning*, E. Alonso, D. Kudenko, and D. Kazakov, Eds. *Lecture Notes In Computer Science*, vol. 2636. Springer-Verlag, Berlin, Heidelberg, 174-186.
- 21) Pinar Saygin, A., Cicekli, I., and Akman, V. 2000. Turing Test: 50 Years Later. *Minds Mach.* 10, 4 (Nov. 2000), 463-518.
- 22) Namatame, A., Kurihara, S., and Nakashima, H. 2007 *Emergent Intelligence of Networked Agents*. 1st. Springer Publishing Company, Incorporated.
- 23) Singh, M. P. 1991. Towards a formal theory of communication for multi-agent systems. In *Proceedings of the 12th international Joint Conference on Artificial intelligence - Volume 1* (Sydney, New South Wales, Australia, August 24 - 30, 1991). J. Mylopoulos and R. Reiter, Eds. *International Joint Conference On Artificial Intelligence*. Morgan Kaufmann Publishers,

San Francisco, CA, 69-74.

- 24) Ossowski, S., García-Serrano, A., and Cuenca, J. 1998. From Theory to Practice in Multiagent System Design: The Case of Structural Cooperation. In Proceedings of the 22nd Annual German Conference on Artificial intelligence: Advances in Artificial intelligence (September 15 - 17, 1998). O. Herzog and A. Günter, Eds. Lecture Notes In Computer Science, vol. 1504. Springer-Verlag, London, 105-116.
- 25) Weyns, D., Helleboogh, A., Holvoet, T., and Schumacher, M. 2009. The agent environment in multi-agent systems: A middleware perspective. *Multiagent Grid Syst.* 5, 1 (Jan. 2009), 93-108.
- 26) 26) Lin, Y., Huang, M., and Chen, J. J. 2006. Agent-based unit testing environment for extreme programming. *J. Comp. Methods in Sci. and Eng.* 6, 5,6 Supplement 1 (Apr. 2006), 1-8.
- 27) Modi, P. J., Mancoridis, S., Mongan, W. M., Regli, W., and Mayk, I. 2006. Towards a reference model for agent-based systems. In Proceedings of the Fifth international Joint Conference on Autonomous Agents and Multiagent Systems (Hakodate, Japan, May 08 - 12, 2006). AAMAS '06. ACM, New York, NY, 1475-1482.
- 28) Bordini, R. H., Dastani, M., and Winikoff, M. 2007. Current issues in multi-agent systems development. In Proceedings of the 7th international Conference on Engineering Societies in the Agents World VII (Dublin, Ireland, September 06 - 08, 2006). G. M. O'Hare, M. J. O'Grady, A. Ricci, and O. Dikenelli, Eds. Lecture Notes In Computer Science. Springer-Verlag, Berlin, Heidelberg, 38-61.
- 29) Dastani, M. and Gomez-sanz, J. J. 2005. Programming multi-agent systems. *Knowl. Eng. Rev.* 20, 2 (Jun. 2005), 151-164.

- 30) Chmiel, K., Gawinecki, M., Kaczmarek, P., Szymczak, M., and Paprzycki, M. 2005. Efficiency of JADE agent platform. *Sci. Program.* 13, 2 (Apr. 2005), 159-172.
- 31) Viroli, M. and Ricci, A. 2004. Instructions-Based Semantics of Agent Mediated Interaction. In *Proceedings of the Third international Joint Conference on Autonomous Agents and Multiagent Systems - Volume 1* (New York, New York, July 19 - 23, 2004). International Conference on Autonomous Agents. IEEE Computer Society, Washington, DC, 102-109.
- 32) Averbukh, V. L. 2001. Visualization Metaphors. *Program. Comput. Softw.* 27, 5 (Sep. 2001), 227-237.
- 33) Averbukh, V., Bakhterev, M., Baydalin, A., Ismagilov, D., and Trushenkova, P. 2007. Interface and visualization metaphors. In *Proceedings of the 12th international Conference on Human-Computer interaction: interaction Platforms and Techniques* (Beijing, China). J. A. Jacko, Ed. Lecture Notes In Computer Science. Springer-Verlag, Berlin, Heidelberg, 13-22.
- 34) Kay, A. 2002. Graphical user interfaces. In *Software Pioneers: Contributions To Software Engineering*, M. Broy and E. Denert, Eds. Springer-Verlag New York, New York, NY, 230-231.
- 35) Greysen, K. R. 2007 *Pedagogical Agent Design: a Grounded Theory Approach*. Doctoral Thesis. UMI Order Number: AAI3295507., University of New Mexico.
- 36) Marsella, S.; Johnson, W. L.; and LaBore, C. 2000. Interactive Pedagogical Drama. In *Proceedings of the Fourth International Conference on Autonomous Agents*, 301–308. New York: Association of Computing Machinery.
- 37) Seow, S. C., Wixon, D., MacKenzie, S., Jacucci, G., Morrison, A., and

Wilson, A. 2009. Multitouch and surface computing. In Proceedings of the 27th international Conference Extended Abstracts on Human Factors in Computing Systems (Boston, MA, USA, April 04 - 09, 2009). CHI '09. ACM, New York, NY, 4767-4770.

- 38) Mark, D. and LaMarche, J. 2009 Beginning Iphone 3 Development: Exploring the Iphone SDK. Apress.

## Публикации

- 39) Дернов Г.С. Использование агентного подхода для разработки обучающей среды как средство обеспечения активного дидактического процесса. // Новые образовательные технологии в вузе (НОТВ-2010): сборник материалов седьмой международной научно-методической конференции, 8 – 10 февраля 2010 года. В 2-х частях. Часть 2. – Екатеринбург: ГОУ ВПО «УГТУ-УПИ имени первого Президента России Б.Н.Ельцина», 2010. Стр. 224-227.
- 40) Дернов Г.С. Использование агентного подхода для разработки обучающей среды как средство обеспечения активного дидактического процесса / под общ. ред. А.Я. Олейникова, Г.Н. Чусавитиной, Л.З. Давлеткиреевой // Управление информационными ресурсами образовательных, научных и производственных организаций: материалы Всероссийской научной школы для молодежи. – Магнитогорск, 2009. Стр. 32-33.
- 41) Авербух В.Л., Дернов Г.С. Использование активных информационных агентов в обучающих средах // Информационно-математические технологии в экономике, технике и образовании: материалы третьей международной конференции. – Екатеринбург, 2008.