

Электронные обучающие системы с использованием интеллектуальных технологий

Рассматриваются основные этапы развития электронных систем обучения. Описаны предпосылки появления интеллектуальных обучающих систем, содержание современных интеллектуальных технологий. Показаны перспективы развития интеллектуальных обучающих систем с применением многократно используемых компонентов (типовых технических решений). Представлена архитектура интеллектуальной обучающей системы с многократно используемыми компонентами. Показаны демо-примеры.

Ключевые слова: электронные системы обучения, интеллектуальная обучающая система, многократно используемые компоненты, система управления базой знаний, интеллектуальные технологии.

E-LEARNING SYSTEMS USING INTELLECTUAL TECHNIQUES

The article covers key stages of e-learning systems development. We try and describe reasons for introduction of intellectual education systems, content of modern intellectual techniques. We also present perspectives for development of intellectual education systems using repeatedly employed components (the typical technical solutions). We present the design of an intellectual education system. Demo-examples are offered as well.

Keywords: e-learning systems, intellectual education system, repeatedly employed components, system of management of knowledge system, intellectual technologies (smart technologies).

Введение

Одной из важных областей деятельности современного общества является образование. Это связано с необходимостью перехода на инновационный путь развития. Классические подходы к образованию – обучение на всю жизнь – не позволяют готовить современных специалистов из-за большой динамики процессов, происходящих в обществе. В таких условиях работники всех уровней должны постоянно обновлять свои знания. Для успешного получения новых знаний необходимы электронные обучающие системы на основе передовых технологий. Решению этой задачи посвящены работы многих специалистов в области информационных технологий [1–6]. В статье показаны предпосылки появления электронных обучающих систем (ЭОС), их использование и развитие на

основе современных информационных технологий. Рассмотрен один из перспективных подходов к созданию современных ЭОС на основе многократно используемых компонентов (типовых технических решений). Показана архитектура интеллектуальной обучающей системы с многократно используемыми компонентами в виде набора модулей системы управления базами знаний.

1. Появление электронных обучающих систем

Передача знаний в ходе обучения являлась важной составляющей деятельности людей в ходе развития цивилизации. Изначально это были различные вспомогательные средства: от простейших приспособлений в древности до механических и электрических систем в последние столетия. Создание и

развитие вычислительных средств ознаменовало появление электронных систем обучения. Первые шаги по применению вычислительной техники в образовании были сделаны в конце 1950-х гг. Исследования в области применения и совершенствования ЭОС начались во всех развитых странах. В своем развитии ЭОС прошли ряд этапов.

Первый этап связан с исследованиями возможностей создания обучающих систем. Эти исследования пришлось на 1950–1960 гг. Уже в 1954 г. профессор Б.Ф. Скиннер выдвинул идею, получившую название программированного обучения [7]. Суть идеи заключалась в необходимости повысить эффективность управления учебным процессом путем построения его в полном соответствии с психологическими знаниями о нем, что фактически означает внедрение ки-

Работа выполнена при финансовой поддержке РФФИ (проект № 13-07-00917а).



Василий Михайлович Трембач,
к.т.н. профессор кафедры
«Прикладная информатика
в экономике»
Тел.: 8 (495) 442-80-98
Эл. почта: trembach@yandex.ru
Московский государственный
университет экономики, статистики
и информатики (МЭСИ)
www.mesi.ru

Vasily M. Trembatch
Candidate of Technical Sciences,
Professor at the Department "Applied
informatics in economics"
Tel.: 8 (495) 442-80-98
E-mail: trembach@yandex.ru
Moscow state university of economics,
statistics and informatics (MESI)
www.mesi.ru

бернетики в практику обучения [8]. В первые годы одним из основных признаков программированного обучения была автоматизация процесса обучения [1, 3, 9], которая началась с использования обучающих и контролирующих устройств различного типа. Эти устройства широко применялись в 1960–1970-е гг. [10], но из-за ограниченных возможностей не обеспечивали достаточной эффективности и адекватности результатов контроля реальному уровню знаний обучаемого. Применение таких устройств [5] способствовало обучению разным навыкам, а также реализации простейших методов контроля, в основном выборочного типа. Одновременно начали развиваться идеи искусственного интеллекта.

При создании первых прототипов автоматизированных обучающих систем (АОС) стало ясно, насколько сложными являются задачи представления предметных знаний, организации обратной связи с обучаемым (в том числе, полноценного диалога, для которого явно не хватало лингвистических знаний). Во многих учебных заведениях велись работы по созданию АОС. Каждый из этих проектов разрабатывался в соответствии с конкретными условиями учебного заведения. В итоге созданные АОС [11], как правило, оказывались несовместимыми между собой в отношении как технического, так и программного обеспечения, что затрудняло обмен учебными курсами и модернизацию АОС.

Разработка АОС активно велась и в зарубежных странах [11]. Для американских систем того времени было характерно стремление к созданию оборудования, специально предназначенного для целей обучения. Велись разработки крупными фирмами, рядом университетов, колледжей, военных и промышленных центров обучения. В основе АОС применялись ЭВМ с различным быстродействием и памятью. Рабочие места учащихся связывались с ЭВМ кабельными системами и спутниковыми каналами связи. С начала 1980-х гг. наметилась тенденция к замене больших ЭВМ общего пользования персональ-

ми ЭВМ и переходу к индивидуальным АОС [5]. С 1970-х гг. в Великобритании интенсивно внедряется вычислительная техника в средние школы, колледжи, университеты. В целом уделялось должное внимание разработке программного и учебно-методического обеспечения АОС и обмену учебными курсами между учебными заведениями. Во Франции в 1979 г. Министерством образования совместно с Министерством промышленности для образовательной системы был разработан план внедрения персональных ЭВМ в обучение. В соответствии с этим планом было осуществлено массовое оснащение лицеев и вузов персональными ЭВМ (программа «10 000 ЭВМ»), создан Центр по распространению дидактических материалов для АОС; составлены многочисленные учебные курсы и проведено массовое обучение преподавателей. В те же годы происходило широкое использование АОС в ФРГ, Нидерландах, Японии и других странах.

2. Интеллектуализация электронных обучающих систем

В 1970 г. Дж. Карбонелл сформулировал общее представление об интеллектуальных обучающих системах (ИОС) [12]. Он представил свою систему SCHOLAR, на примере которой была продемонстрирована эффективность использования методов искусственного интеллекта (ИИ) в такой области, как обучение [1, 12]. Если началом исследований в области ИИ принято считать 1956 г., когда Ньюэлл и Саймон приступили к исследованиям «сложных процессов обработки информации» в Технологическом институте Карнеги, то 1970-й можно считать годом рождения нового научного направления, появившегося на стыке программированного обучения и ИИ [2, 9, 12]. ИОС позволяли выбирать или формировать обучающие воздействия в зависимости от целей обучения и с учетом знаний учащегося.

Второй этап развития электронных систем обучения включал период с начала 1970-х до середины 1980-х гг. В эти годы в истории ис-

куственного интеллекта наступили трудные времена, соответственно, и идея создания интеллектуальных обучающих систем фактически потерпела временное фиаско, что нашло свое отражение в деградации понятия автоматизированного обучения. Автоматизированными обучающими системами начали называть любые программы, предназначенные для информационной или функциональной поддержки процесса обучения: тесты, электронные учебники, лабораторные практикумы и т.п. Но несмотря на ослабление требований к обучающим системам, продолжались исследования возможностей использования, при создании АОС, идей и методов представления знаний, разработанных к тому времени в области ИИ. Реальные исследовательские и коммерческие ИОС появились уже в 1980-е гг. и были нацелены на диагностику, отладку и коррекцию поведения обучаемого. Такие системы уже не только диагностируют и указывают студенту его ошибки, но и анализируют их причины, строят гипотезы, правила и планы исправления ошибок, дают советы, исходя из предварительно определенных стратегий обучения и имеющейся модели обучаемого [1, 12].

Третий этап начался во второй половине 1980-х и завершился в 1990-е гг. В этот период просматривались две основные тенденции:

- широкое распространение персональных компьютеров (ПК) и развитие вычислительных сетей ориентирует обучающие системы на работу в сети с использованием общепринятых стандартов представления и передачи данных;

- возросшие аппаратные возможности привели к тому, что одним из основных направлений развития обучающих систем стало применение в них новых компьютерных технологий (в первую очередь, гипертекста, мультимедиа, технологий ИИ).

Повальное увлечение новомодными технологиями отодвинуло на второй план содержательную и методическую составляющие обучающих систем.

В середине 1980-х гг. стало ясно, что интеллектуализация

обучающих систем в первую очередь связана с практическим использованием, при их разработке и реализации, методов и средств, созданных в рамках исследований по экспертным системам. Это, в свою очередь, вызвало к жизни серьезные исследования по моделям объяснения в обучающих системах, с одной стороны, и интеллектуальным технологиям формирования моделей предметной области, стратегий обучения и оценки знаний обучаемых, на основе более сложных моделей самих обучаемых, с другой стороны. Сложившаяся ситуация позволила говорить об адаптирующихся обучающих системах, которые могли в зависимости от параметров обучаемого и результатов контроля знаний обучаемого генерировать новые последовательности управляющих воздействий. В итоге на ранних стадиях развития ИОС подходы, разработанные в рамках ИИ, использовались лишь для представления знаний из предметной области. А с середины 1980-х гг. в состав ИОС стала, в явном виде, включаться информация об обучаемом (в частности, модель идеального обучаемого) и стратегии обучения [1, 12].

С 1990-х гг. берет начало современный, четвертый этап. Его содержание тесно связано с развитием сети Интернет. По сравнению с локальными обучающими системами, в распределенных происходит качественное изменение функциональных возможностей, благодаря объединению сетевых ресурсов для решения возникающих перед обучающей системой задач. При работе через вычислительную сеть общение между обучаемыми и преподавателем стало более интенсивным, что позволяло преподавателю постоянно контролировать состояние процесса обучения (используя компьютерные средства контроля), а обучаемому консультироваться в режиме онлайн или по электронной почте.

Развитие сетевых технологий и результаты исследований в области ИИ дали возможность для создания перспективных обучающих систем, которые позволяют адаптировать учебный процесс к конкретному

обучаемому. Уже в 1990-е гг. при построении ИОС начали применять агентно-ориентированную технологию [1, 4]. Сформировалось и в дальнейшем укрепляется понимание того, что будущие обучающие системы будут создаваться с использованием возможностей вычислительных сетей, средств телекоммуникации и интеллектуальных технологий. Но простой перенос в информационную среду рассмотренных подходов к организации образовательного процесса не обеспечивает требуемую эффективность в получении знаний. Решение данной проблемы связано с реализацией компетентного подхода в образовании, заключающегося в переносе акцента с содержания обучения на его результат [2, 13–16].

Использование основных положений компетентного подхода в непрерывном образовании означает формирование разнообразных образовательных программ для различных категорий обучающихся по различным видам и формам обучения. Данная задача может быть решена за счет введения адаптивности в процесс обучения. Основная проблема заключается в планировании и реализации образовательной траектории обучающегося к цели на основе управляемого усвоения им учебного материала в обучающей системе. Индивидуальную образовательную траекторию обучающегося [5, 17] можно определить как персональный путь реализации личностного потенциала обучающегося в образовательном процессе. Согласно психолого-педагогической литературе ценность индивидуальной образовательной траектории обучаемого заключается в том, что она позволяет на основе оперативно регулируемой самооценки, активного стремления к совершенствованию собственных знаний и умений пополнить знания при проектировании своей учебной деятельности с целью отработки методов и техники самостоятельной работы в различных формах учебно-познавательной деятельности. При этом очень важно наличие у каждого обучаемого своей индивидуальной задачи по

проектированию индивидуальной образовательной траектории, что способствует повышению личностного образовательного роста обучаемого.

Достижению успеха в этом направлении способствует то, что в современных системах электронного обучения реализуется концепция учебных объектов. Учебный объект – относительно молодое понятие в образовательных технологиях. По определению Комитета по стандартам обучающихся технологий IEEE (IEEE LTSC), учебный объект – это «любой объект, цифровой или нецифровой, который может использоваться многократно, на который можно делать ссылки при использовании соответствующей обучающей технологии». Главными характеристиками учебного объекта являются возможность многократного использования и разметка метаданными.

Учебные объекты делают возможным создание независимых компонентов образовательного контента, которые обеспечивают реализацию образовательных целей. Существуют многие подходы к их классификации, структуре, описаниям составных элементов. IMS и SCORM обеспечивают детальные стандарты и рекомендации для учебных объектов. Стандарт SCORM [18, 19] определяет структуру учебных материалов и интерфейс среды выполнения. Благодаря этому, учебные объекты могут быть использованы в различных системах электронного дистанционного образования. SCORM описывает эту структуру с помощью нескольких основных принципов, спецификаций и стандартов, основываясь при этом на других уже созданных спецификациях и стандартах электронного и дистанционного образования.

Составной частью SCORM являются метаданные учебных объектов (Learning Object Metadata, LOM). Цель этого стандарта – облегчение поиска, рассмотрения, оценки и использования учебных объектов для учеников, учителей или автоматических программных процессов. Определяя общую концептуальную схему данных, дан-

ный стандарт обеспечивает связывание учебных объектов.

Стандарты для метаданных определяют минимальный набор атрибутов, необходимый для организации, определения местонахождения и оценки учебных объектов. Значимыми атрибутами учебных объектов являются:

- тип объекта;
- имя автора объекта;
- имя владельца объекта;
- сроки распространения;
- формат объекта.

Использование метаданных учебных объектов является центральным звеном в ИОС, которые могут учитывать индивидуальные требования к обучению.

3. Особенности создания современных интеллектуальных обучающих систем

Современные ИОС, разработанные на основе учебных объектов, используют мета-данные, которые хранятся в базах знаний. К настоящему времени нет общепризнанных определений понятиям «база данных» и «база знаний». Автор принимает точку зрения, изложенную в работе [20], где отмечается, что технических различий между терминами «база знаний» и «база данных» нет. Это объясняется тем, что многофункциональные (расширенные) системы управления базами данных, такие как управление объектно-ориентированными, активными и дедуктивными базами данных, поддерживают некоторый дедуктивный и не дедуктивный механизмы выводов и средства структурирования, аналогичные базам знаний. Разница в значении этих двух терминов, если она имеется, в основном в том, в какой степени системы поддерживают представление, структурирование и возможность вывода.

В работе [20] представлена архитектура системы управления базами знаний (СУБЗ). С 1985 по 1995 г. в университете Торонто (Канада) эта СУБЗ разрабатывалась в проекте KBMS (Knowledge Base Management System) [20, 21]. Основной целью этого проекта было создание универсальной архитек-

туры СУБЗ, предназначенной для развивающихся компьютерных приложений. В основу разработанной СУБЗ положена расширяемая, многоуровневая архитектура. Расширяемая архитектура позволяет СУБЗ работать как с механизмами вывода общего назначения, так и со специальными механизмами вывода. Специальные механизмы вывода, например механизм пространственного мышления или механизм на основе доказательной аргументации, встраиваются в зависимости от потребностей специальных приложений, в то время как механизмы вывода общего назначения являются одинаковыми для всех приложений. Многоуровневая архитектура поддерживает проектирование кода, основанное на повышении уровня абстракции, что позволяет разбивать общую задачу проектирования СУБЗ на несколько подзадач. В такой архитектуре использован стандартный интерфейс для каждого уровня и его компонентов, что позволяет многократно их использовать в различных СУБЗ. Разработанная архитектура системы управления базой знаний (рис. 1) включает три уровня [20]:

- уровень интерфейсов, который предлагает различные виды пользовательских интерфейсов;
- логический уровень, который выполняет примитивные операции по извлечению знаний и обновлению базы знаний;
- физический уровень, который управляет структурами данных для хранения баз знаний, различных показателей и другой вспомогательной информации.

Уровень интерфейсов предлагает множество пользовательских служб баз знаний, включая гипертекстовый интерфейс для оперативного взаимодействия с пользователем и интерфейс языков программирования (PL), который поддерживает выполнение прикладных программ, включающих операции базы знаний. Кроме того, уровень интерфейсов может включать инструменты системы управления базой знаний для сбора знаний, проверки базы знаний, проверки ограничений, развития и обмена знаниями [22, 23]. Службы уровня интерфейсов соеди-

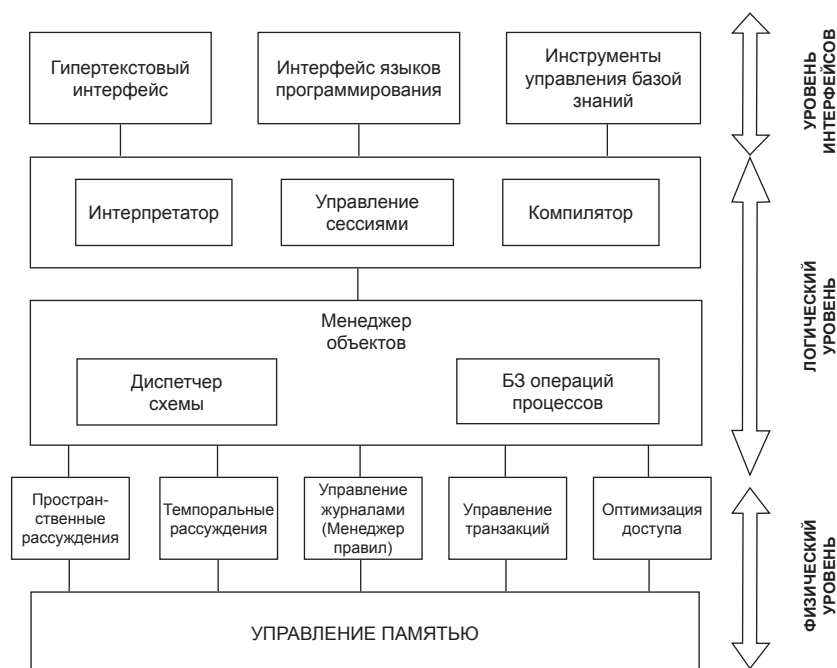


Рис. 1. Общая архитектура системы управления БЗ

няются с логическим уровнем через интерпретатор языка представления знаний, управление сессиями и компилятор.

Логический уровень поддерживает информацию об определениях классов, включая правила и ограничения, и поддерживает примитивные операции базы знаний [24]. Его службы реализованы поверх физического уровня (набора модулей), которые предусматривают функции управления исходными данными: трассировка пути доступа, планирование эффективной обработки запроса и параллельное выполнение протоколов, блоки рассуждений специального назначения для временных, пространственных или других типов обоснования (логик), а также компонент управления правилами, который поддерживает дедуктивный вывод и проверку ограничений.

Физический уровень отвечает за управление: структурами данных, находящимися на диске, и на которых база знаний сохранена; индексами, поддерживаемыми архитектурой; политикой кэширования и т.д.

Функциональность нижней части этого уровня обеспечена ядром хранения баз данных, таким же, как разработанные для объектно-ориентированных и вложенных реляционных баз данных [25, 26, 27].

Созданная в рамках проекта KBMS архитектура системы управления знаниями, используемые технологии и подходы к представлению и использованию знаний нашли применения во многих приложениях ИИ, использующих БЗ [21].

При решении задач реальной сложности система управления БЗ должна иметь возможности для концептуального представления действительности, хранения и использования высокоструктурированных знаний. Наличие в архитектуре СУБЗ проекта KBMS [21] на уровне интерфейсов модуля инструментов управления БЗ, интерфейса языков программирования позволяет иметь блоки управления БЗ для решения различных задач. Но современные приложения являются сложными, ориентированными на меняющиеся задачи и методы их решения, что требует больших затрат на их актуализацию и даже перепроектирование интеллектуальной системы. Проектирование и перепроектирование – это ресурсоемкие мероприятия, так как в каждом приложении необходимо создание своих, уникальных модулей инструментов управления БЗ. Для уменьшения затрат при создании новых систем, основанных на знаниях (СОЗ), к которым относится и ИОС, необходимо в СУБЗ наличие готовых оболочек

различных инструментов не только для управления БЗ, но и решения прикладных задач.

Для непрерывного образования в авиакосмической области необходимо решать ряд задач, связанных с созданием и использованием индивидуальной среды обучения и тестирования, с актуализацией знаний об учебных объектах, валидностью БЗ. Например, в летной практике имеется множество задач формирования навыков и знаний, которые меняются быстро. Заготовки решений для этих задач могут составить основу СУБЗ интеллектуальных систем в области непрерывного образования пилотов. Важным моментом для СУБЗ является метод представления знаний, позволяющий описывать метаданные об учебных объектах и обеспечивающий решение различных задач. Основными из этих задач являются:

- создание и использование индивидуальной среды обучения, тестирования,
- формирование последовательности учебных, тестовых объектов,
- актуализация знаний с помощью процедур обучения.
- проверка валидности БЗ на основе их моделирования сетями Петри.

При создании ИОС, использующих метаданные учебных объектов (знания об учебных объектах), одной из сложных проблем является выбор адекватного метода их представления. Для применения индивидуальных траекторий обучения предлагается интегрированный метод, в основе которого заложены возможности представления структуры предметной области и взаимодействия сущностей предметной области друг с другом. Интегрированный метод представления знаний позволяет задавать предметную область множеством описаний концептов. Описания концептов представляются как множества вершин и взвешенных связей между ними. Каждая вершина описывается следующими атрибутами представления сущности [5, 14, 28, 29]:

- имя;
- предусловие;
- постусловие;

- имена концептов нижнего уровня;
- имена концептов верхнего уровня;
- имена концептов-ассоциаций.

Современная ИОС является сложной распределенной системой. Компонентами такой распределенной системы являются множество субъектов учебного процесса, которые обладают характерными для человека сложным поведением, интеллектом и индивидуальными средствами коммуникации, что делает неэффективным применение традиционных формальных методов для их описания. Поэтому в настоящее время широко используется агентно-ориентированный подход при создании приложений реальной сложности.

4. Архитектура интеллектуальной обучающей системы

Для непрерывного образования можно выделить задачи, которые решаются ИОС на всех этапах формирования компетенций обучающихся. Такими задачами являются:

- регистрация пользователей;
- получение доступа к персональной среде;
- просмотр базы знаний с обеспечением целостности данных, исключением ошибок ввода, облегчением ввода данных, автоматизацией обработки описаний на множестве объектов и поиском;
- просмотр, наполнение и редактирование репозитория с широкими возможностями в оформлении учебного материала, большим набором мультимедийного наполнения, простотой и удобством, как создания новых учебных статей, так и их редактирования, с обеспечением коллективного доступа, наличием механизма ревизии описаний;
- ввод текущих оценок компетенции;
- оценка уровня знаний;
- контроль получения знаний;
- формирование индивидуальных траекторий – планирование индивидуальной программы обучения;
- реализация индивидуальных программ обучения с использованием индивидуальной среды обучения.

Для решения выделенных задач в интеллектуальной обучающей системе на основе агентно-ориентированного подхода можно выделить следующие компоненты:

- интерфейс обучаемого;
- модуль оценки знаний обучаемого;
- подсистема формирования индивидуальных планов обучения;
- система управления базой знаний;
- индивидуальная среда обучения;
- репозиторий учебных объектов.

Такая ИОС ориентирована на индивидуальную работу с обучающимися. Интерфейс обучаемого является по своей сути аппаратно-программной сущностью, обеспечивающей обучаемому возможность работы со всеми имеющимися в системе сервисами. Этот модуль позволяет формировать и хранить требуемые компетенции, текущие компетенции обучаемого и сформированные индивидуальные программы обучения. Модуль оценки знаний дает обучаемому возможности определять свой текущий уровень компетенций, контролировать процесс отработки индивидуальной программы обучения. Подсистема формирования индивидуальных программ обучения осуществляет планирование последовательности учебных объектов в зависимости от требуемой компетенции и имеющихся у обучаемого знаний, умений, навыков. Система управления базой знаний обеспечивает функции системы управления базами данных и функции, связанные с формированием и использованием знаний в работе с обучаемым. Индивидуальная среда обучения должна обеспечить возможность работы со всеми учебными объектами в рамках спланированной последовательности. Репозиторий учебных объектов должен обеспечить создание, хранение и использование учебных объектов различной природы.

Разработка современных ИОС связана с рядом проблем, которые сдерживают эффективные технологии проектирования интеллектуальных систем [30]. Для их преодоления необходимы определенные условия и наработки. Одним из

условий должно стать создание эффективной технологии проектирования ИОС. В работе [30] для решения этой задачи предлагается методика компонентного проектирования, которая является фактором зрелости любых технологий и которая основывается на постоянно расширяемых библиотеках многократно используемых компонентов (типовых технических решений). Для этого требуется решение многих вопросов, некоторые из них представлены ниже:

- обеспечение совместимости (интегрируемости) компонентов интеллектуальных систем на основе унификации представления этих компонентов;
- создание библиотек многократно используемых (типовых) компонентов интеллектуальных систем и уточнение типологии таких компонентов (предметные онтологии, многократно используемые фрагменты баз знаний, машины вывода, интерфейсные компоненты и т.д.);
- создание средств компьютерной поддержки синтеза интеллектуальных систем из имеющихся компонентов и некоторые другие [30].

Требуется также независимость процесса обновления базы знаний интеллектуальной системы от процесса обновления моделей, а также методов обработки знаний от процесса обновления средств технической реализации.

Необходимость решения этих вопросов связана с тем, что в каждой предметной области множество задач и необходимых решений может несколько изменяться. Так, например, в авиакосмической области для решения в ИОС задач непрерывного образования ее СУБЗ необходимы следующие технические решения:

- интегрированный метод представления знаний;
- блок для формирования и реализации планов обучения;
- блок для обучения на собственном опыте и успешных решениях;
- блок моделирования БЗ сетями Петри;
- блок оценки знаний обучаемых.

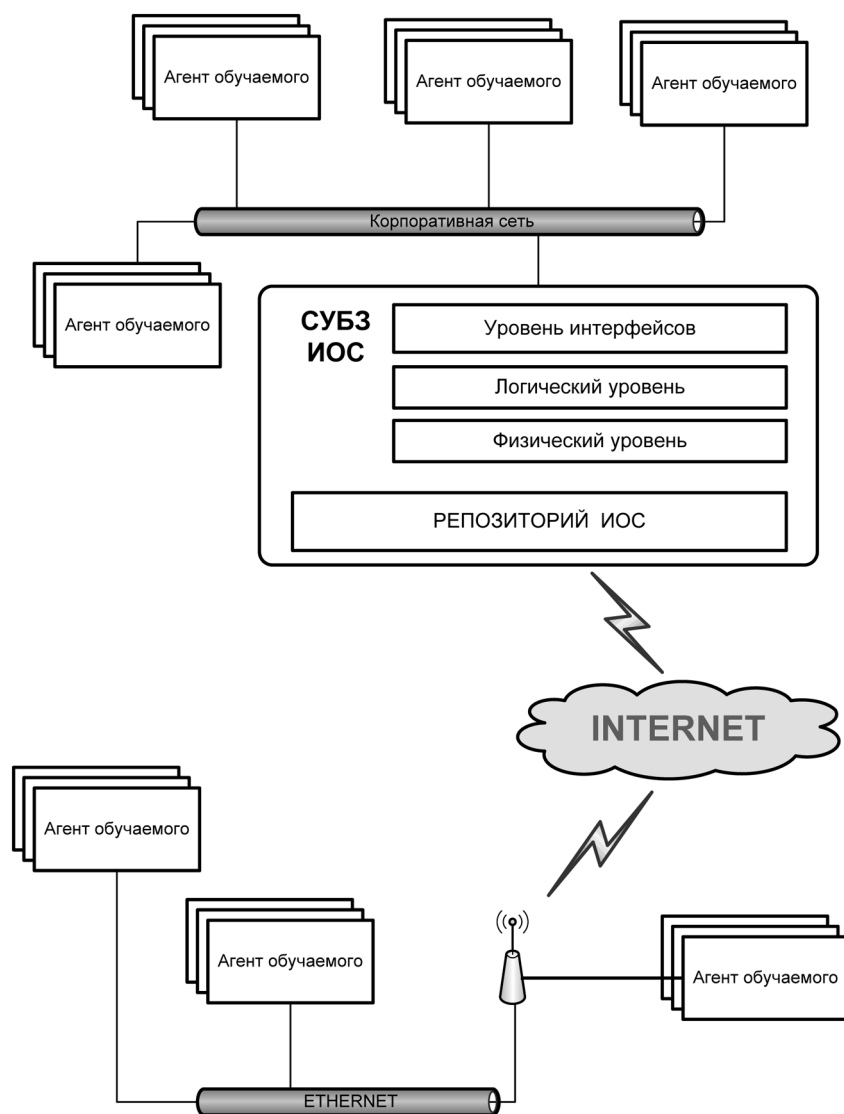


Рис. 2. Архитектура ИОС

Эти решения являются типовыми и могут использоваться для каждого обучаемого и составлять библиотеку многократно используемых (типовых) компонентов интеллектуальных обучающих систем.

Интегрированный метод представления знаний дает возможность представить действительность в виде множества сущностей и связей между ними. Это позволяет описывать таксономию предметной области и процессы, порождаемые сущностями в этой области. Такое представление позволяет решать задачи непрерывного образования

Блок для формирования и реализации планов обучения позволяет использовать персональную среду обучения, которая обеспечивает формирование новых компетенций

каждым обучаемым с учетом его текущих знаний, умений, навыков.

Работу с актуальными знаниями, их соответствие действительности обеспечивает блок для обучения на собственном опыте и успешных решениях. В этом блоке реализуются методы машинного обучения.

Большие объемы баз знаний современных интеллектуальных систем, и в том числе ИОС, повышают требования к корректности содержимого БЗ. Одним из подходов к валидации баз знаний рассматривается использование сетей Петри. Предлагаемые в [5] методы основаны на композиции сети Петри из множества функциональных подсетей, что делает их пригодными для моделирования баз знаний ИОС.

Для полноценного функционирования персональной среды обучаемого необходим блок оценки знаний. Он обеспечивает мониторинг отработки индивидуальной траектории обучения.

Архитектура глобальной ИОС, обеспечивающая реализацию рассмотренных решений, показана на рис. 2.

Данная архитектура включает ядро ИОС – систему управления базами знаний глобальной ИОС. На уровне интерфейсов СУБЗ находится блок формирования и поддержки агентов обучаемого. С помощью многократно используемых компонентов (типовых технических решений) формируется агент для каждого обучаемого. В сформированном агенте обучаемого используются следующие компоненты: интерфейс обучаемого с персональной средой; формирователь (планировщик) индивидуальной траектории; модуль (агент) для оценки знаний обучаемого на всех этапах формирования компетенций; модуль персональной среды обучения для реализации индивидуальной траектории обучения; модуль формирования репозитория персональной среды обучения и поддержания его в актуальном состоянии.

Для работы СУБЗ ИОС могут потребоваться многократно используемые компоненты для решения задач машинного обучения (накопления опыта), актуализации содержимого базы знаний ИОС, распознавания типовых ситуаций в обучении, валидации баз знаний и другие.

5. Демонстрация системы

Использование подходов компонентного проектирования [30] и агентно-ориентированных технологий [4, 5, 29, 31] позволяет создавать ИОС как открытую систему. При разработке компонентов ИОС могут использоваться многие программные средства. При выборе языка разработки рассматривались следующие критерии:

- возможности языка;
- функциональность встроенных библиотек;
- функциональность сторонних библиотек и фреймворков.

Наличие (развитие) сторонних библиотек для языков программирования и требования к современным приложениям не позволяют рассматривать язык программирования в отрыве от того, насколько он распространен, как сильно развито сообщество разработчиков, наличия документации, поддержки, сторонних библиотек и готовых приложений для выполнения типовых задач.

При анализе языков программирования необходимо рассматривать доступные библиотеки и фреймворки сторонних разработчиков с поддержкой рассматриваемого языка, которые могут быть использованы для реализации типовых системных функций прототипа агента для формирования траектории обучения. Доступные библиотеки и фреймворки сторонних разработчиков должны иметь поддержку следующих функций:

- сериализации данных в базе данных;
- авторизации пользователей;
- инструментария и библиотеки времени выполнения, для реализации пользовательского интерфейса, например веб-интерфейса;
- реализации коммуникации между компонентами системы.

Перечисленные функции являются типовыми для большинства приложений, требуют высокой квалификации исполнителей для реализации и достаточно трудозатратны. Поэтому целесообразно ис-

пользовать сторонние библиотеки и фреймворки для реализации типовой базовой функциональности.

Для реализации прототипа агента, формирующего траектории обучения, выбран сценарный (динамический) язык программирования. Динамический язык является удобным средством для быстрой разработки приложения. Использование сценарного языка программирования позволяет значительно (более чем в два раза) уменьшить время, затрачиваемое на проектирование, написание и тестирование программы [32], что является одним из ключевых факторов при разработке прототипа. Сценарный (динамический) язык программирования позволяет определять типы данных и осуществлять синтаксический анализ и компиляцию на этапе выполнения программы [33]. К динамическим языкам относятся Perl, Tcl, Python, PHP, Ruby, Smalltalk, JavaScript, а также Visual Basic. Для реализации прототипа агента, формирующего траектории обучения, выбран язык Python.

Стандартная библиотека Python содержит множество модулей, реализующих самые разные функции, которые необходимы для разработки приложений. Они в полной мере задействованы при разработке прототипа агента для формирования индивидуальной траектории обучения. Важной частью веб-приложения прототипа агента для формирования траектории обучения

является простая и функциональная схема адресных ссылок внутри приложения. Для реализации схемы адресных ссылок в фреймворке для быстрой разработки веб-приложений – Django используется модуль URLconf, который позволяет создать таблицу содержимого приложения с простым соответствием между адресной строкой (URL) и функцией Python, которая реализует генерацию страницы пользовательского интерфейса в веб-приложении. Пример схемы адресных ссылок для прототипа агента формирования индивидуальной траектории обучения показан на рис. 3.

В приведенном примере на рис. 3 показана таблица, состоящая из двух адресных ссылок веб-приложения для отображения страниц со списком всех студентов, зарегистрированных в системе, и адресной ссылки для отображения индивидуальной карточки студента. Пример исходного кода Python для генерации страницы со списком студентов, с использованием фреймворка Django, показан на рис. 4. В приведенном примере показано использование модели ORM Django для чтения информации о студентах из базы данных (`Student.objects.all()`) и генерации веб-страницы на основе html-шаблона из файла `all_students.html`.

Пример html-шаблона для отображения списка студентов показан на рис. 5. Этот шаблон написан на языке разметки HTML, с исполь-

```
from django.conf.urls import patterns, include, url

urlpatterns = patterns('',
    (r'^students$', 'web.students.views.all_students'),
    (r'^student/(?P<full_name>.*)/$', 'web.students.views.student_card'),
)
```

Рис. 3. Схема адресных ссылок

```
from django.shortcuts import render_to_response
from web.students.models import Student

def all_students(request):
    students = Student.objects.all()
    return render_to_response('all_students.html', {'students_list': students})
```

Рис. 4. Функция Python для генерации веб-страницы пользовательского интерфейса


```
{% extends "base.html" %}

{% block content %}
<h1>Список студентов</h1>

{% for s in students_list %}
  <p>{{ s.full_name }}
    <a href="{% url 'web.students.views.student_card'
s.full_name %}">[карточка]</a></p>
{% endfor %}

{% endblock %}
```

Рис. 5. Пример шаблона веб-страницы пользовательского интерфейса для отображения списка студентов

зованием ключевых выражений фреймворка Django, которые реализуют возможность добавления динамического содержимого в статичную страницу шаблона. Например, генерация списка с именами студентов реализована как цикл `for` по элементам списка Python с объектами, содержащими записи из базы данных.

Таким образом, создаются элементы, необходимые для создания пользовательского интерфейса веб-приложения на базе фреймворка Django на языке программирования Python.

Для проверки в приведенных примерах всех элементов вместе и просмотра результата можно использовать встроенный в Django веб-сервер разработки. Для запуска сервера необходимо ввести в командной строке команду `python manage.py runserver`. По умолчанию, страницы веб-приложения, запущенные с использованием веб-сервера разработки в составе Django, будут доступны по адресу

`http://localhost:8000/`. Открывшаяся страница веб-приложения показана на рис. 6. Для этого примера веб-страницы со списком студентов в пользовательском интерфейсе использовался фреймворк Django на языке программирования Python.

Для описания в ИОС метаданных об учебных объектах использовался формат YAML. YAML — это язык сериализации данных в человекочитаемом формате, который вобрал в себя концепции языков программирования, таких как Си, Perl и Python, а также идеи языка разметки XML и формата электронных писем [34, 35].

Аббревиатура YAML – это рекурсивный акроним «YAML Ain't Markup Language», что в переводе означает «YAML не язык разметки», хотя первоначальной версией было выражение «Yet Another Markup Language», т.е. «еще один язык разметки». Эта смена интерпретации была сделана, чтобы подчеркнуть назначение YAML,

как языка, ориентированного на данные, в противовес языкам разметки документов, например HTML и др. [35].

Язык YAML был специально разработан для прямого отображения наиболее широко распространенных типов данных в языках программирования, таких как списки, ассоциативные массивы и значения. Использование характерных отступов в YAML особенно практично для задач, когда язык разметки будет использоваться для просмотра и редактирования пользователем, например в конфигурационных файлах, описании метаданных документов и других структурированных данных с изменяемым форматом.

Метаданные в описании учебного объекта включают следующие атрибуты:

- имя учебного объекта;
- перечисление целевых компетенций;
- перечисление требуемых компетенций.

Метаданные УО могут дополнительно содержать следующие поля:

- перечисление включенных элементов в рассматриваемый УО;
- перечисление элементов, включающих рассматриваемый УО;
- перечисление имен связей для рассматриваемого УО;
- перечисление имен ассоциаций для рассматриваемого УО;
- ресурсы (время, стоимость и прочее).

В формате YAML обязательные поля метаданных УО записываются, как показано на рис. 7.

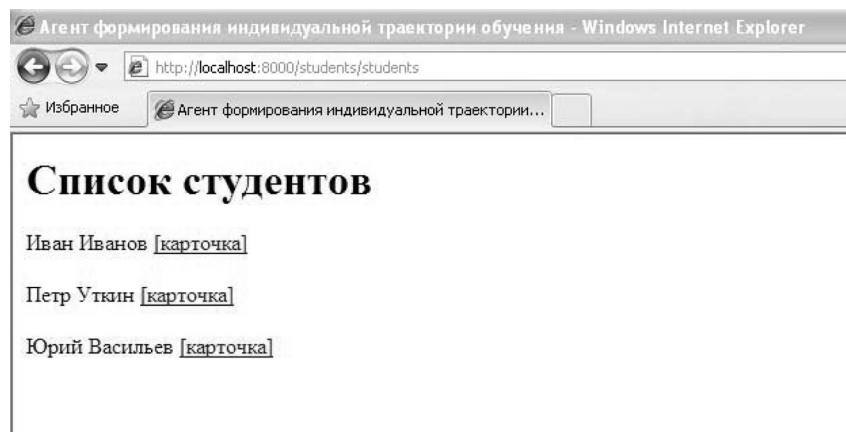


Рис. 6. Веб-страница со списком студентов в пользовательском интерфейсе

```

substances:
- name: <Имя Учебного Объекта 1>
  psus:
    <Постусловние/Целевая компетенция1>: <Значение>
  prus:
    <Предусловие/Требуемая компетенция1>: <Значение>
    <Предусловие/Требуемая компетенция2>: <Значение>
- name: <Имя Учебного Объекта 2>
  psus:
    <Постусловние/Целевая компетенция2>: <Значение>
  prus:
    <Предусловие/Требуемая компетенция3>: <Значение>
    <Предусловие/Требуемая компетенция4>: <Значение>

```

Рис. 7. Структура метаданных об учебных объектах в YAML

```

substances:
- name: "Обучение методам представление знаний"
  psus:
    "Знание методов представления знаний": 1
  prus:
    "Знание продукционных моделей": 1
    "Знание семантических сетей": 1
- name: "Обучение продукционным моделям"
  psus:
    "Знание продукционных моделей": 1
  prus:
- name: "Обучение семантическим сетям"
  psus:
    "Знание семантических сетей": 1
  prus:

```

Рис. 8. Метаданные УО «Обучение методам представления знаний» в YAML

На рис. 8 показан пример описания метаданных учебных объектов из предметной области для обучения методам представления знаний.

Заключение

В развитии современных электронных обучающих систем все больше проявляется их ориентация

на интеллектуализацию. Причиной этой тенденции является динамика процессов во всех областях экономики, что требует использования специалистами современных, актуализированных знаний. Получение знаний по классическим технологиям происходит с запаздыванием и уже не обеспечивает

конкурентного преимущества обучаемым. Поэтому формирование компетенций, соответствующих современным требованиям, возможно только с использованием интеллектуальных обучающих систем, содержащих персональные среды обучения и учитывающих текущие компетенции обучаемого.

Литература

1. *Голенков В.В., Емельянов В.В., Тарасов В.Б.* Виртуальные кафедры и интеллектуальные обучающие системы // *Новости искусственного интеллекта.* – 2001. – № 4.
2. *Рыбина Г.В.* Экспертные системы и инструментальные средства для их разработки: некоторые итоги // *Информационно-измерительные и управляющие системы.* 2013. – Т. 11, № 5. – С. 35–48.
3. *Стефанюк В.Л.* Поведение квазистатической оболочки в изменяющейся нечеткой среде // В кн.: КИИ-94. Национальная конференция с международным участием «Искусственный интеллект 94»: сборник научных трудов: в 2-х т. Т. 1. – Рыбинск 1994. – С. 199–203.
4. *Тарасов В.Б.* От многоагентных систем к интеллектуальным организациям: философия, психология, информатика. – М.: Эдиториал УРСС, 2002. – 352 с.
5. *Трембач В.М.* Системы управления базами эволюционирующих знаний для решения задач непрерывного образования: монография. – М.: МЭСИ, 2013. – 255 с.
6. *Telnov Y.* The Model of Continuous Profession-oriented Learning in the E-environment Based on a Competence Approach and Academic Knowledge Management // 11th European Conference of Knowledge Management Systems, 2–3 Sept, 2010, Porto.

7. Skinner, B.F. The science of learning and art of teaching // Harvard Education Review. 1954. Spring, № 24. – P. 86–97.
8. Талызина Н.Ф. Теоретические проблемы программированного обучения. – М.: Изд-во МГУ, 1969. – 133 с.
9. Рыбина Г.В. Основы построения интеллектуальных систем: учеб. пособие / Г.В. Рыбина. – М.: Финансы и статистика: Инфра-М, 2010. – 432 с.
10. Кибернетика и проблемы обучения: сборник переводов / ред. и предисл. А.И. Берга. – М.: Прогресс, 1970. – 389 с.
11. Педагогическая энциклопедия: http://www.eslovar.info/?grupa=11&id_sl=11
12. Carbonell, J.R. AI in CAI: an Artificial Intelligence Approach to Computer-Aided Instruction // IEEE Transactions on Man-Machine Systems. – 1970. – Vol. MMS-11, № 4.
13. Трёмбач В.М. Основные этапы создания интеллектуальных обучающих систем // Программные продукты и системы, №3, 2012. С. 148–152.
14. Трёмбач В.М. Решение задач управления в организационно-технических системах с использованием эволюционирующих знаний: монография. – М.: МЭСИ, 2010. – 235 с.
15. Tikhomirova, N., Tikhomirov, V., Telnov, Y. and Maksimova, V. Chapter 9. The University's Integrated Knowledge Space in Knowledge Management // In Search of Knowledge Management: Pursuing Primary Principles / edited by A. Green, M. Stankosky, L. Vandergriff. Emerald Group Publishing Limited, 2010. – P. 147–162.
16. Smart education – веление времени [Электронный ресурс] // Российская газета – Прикамье. – 2011. – 2 марта. – Режим доступа: <http://www.media-office.ru/?go=1710346&pass=cf3478cd14cf8574154d30268e8116d3> (дата обращения 25.06.2013).
17. Загорюлько Ю.А. Методологические проблемы построения онтологий для портала научных знаний // Когнитивные исследования: сб. науч. тр. / РАН, Ин-т психологии, Казан. гос. ун-т им., Ассоц. когнитивных исследований; отв. ред. В.Д. Соловьев, Т.В. Черниговская. – М. Ин-т психологии РАН, 2006. – С. 308–317.
18. IMS Global Learning Consortium: <http://www.imsglobal.org/question/> (дата обращения 22.06.2013).
19. SCORM – сборник спецификаций и стандартов для систем дистанционного обучения. – Режим доступа: <http://www.edu.ru/db/portal/e-library/00000053/SCORM-2004.pdf> (дата обращения 22.06.2013).
20. Mylopoulos, J., Chaudhri, V.K., Plexousakis, D., Shrufi, A., Topaloglou, T. Building knowledge base management systems // The VLDB Journal. – 1996. – № 5(4). – P. 238–263.
21. Vinay K. Chaudhri, Jurisica I., Koubarakis M., Plexousakis D., Topaloglou T. The KBMS Project and Beyond. Conceptual Modeling: Foundations and Applications 2009: 466–482.
22. Buchanan, B.G., Wilkins, D.C. (1993) Readings in knowledge acquisition and learning. Morgan Kaufmann, San Mateo, Calif.
23. Neches, R., Fikes, R., Finin, T., Gruber, T., Patil, R., Senator, T., Swartout, W. (1991) Enabling technology for knowledge sharing. AI Mag 12:36–56.
24. Mylopoulos, J., Borgida, A., Jarke, M., Koubarakis, M. (1990) Telos: a language for representing knowledge about information systems. ACM Trans Inf Sys 8:4–362.
25. Biliris, A. (1992) The performance of three database storage structures for managing large objects // In: Proceedings of the ACM SIGMOD International Conference on Management of Data.
26. Carey, M., DeWitt, D., Richardson, J., Shekita, E. (1986) Object and file management in the EXODUS extensible database system // In: Proceedings of the 12th International Conference on Very Large Data Bases. P. 91–100.
27. Paul, H.-B., Schek, H.-J., Scholl, M., Weikum, G., Deppish, U. (1987) Architecture and implementation of a Darmstadt database kernel system // In: Proceedings of the ACM SIGMOD International Conference on Management of Data. – San Francisco, Calif, P. 196–207.
28. Трёмбач В.М. Интеллектуальные технологии для решения задач непрерывного образования // Открытое образование. – 2012. – № 3. – С. 4–11.
29. Трёмбач В.М. Формирование и использование моделей компетенций обучающихся на основе эволюционирующих знаний // Открытое образование. – 2009. – № 6 (77). – С. 12–26.
30. Голенков В.В., Гулякина Н.А. Открытый проект, направленный на создание технологии компонентного проектирования интеллектуальных систем // В кн.: Междунар. научн.-техн. конф. «Открытые семантические технологии проектирования интеллектуальных систем = Open Semantic Technologies for Intelligent Systems (OSTIS-2013)»: материалы III Междунар. научн.-техн. конф. (Минск, 21–23 февраля 2013 г.). – Минск: БГУИР, 2013. – С. 55–77.
31. Рассел, Стюарт, Норвиг, Питер. Искусственный интеллект: современный подход: пер. с англ. – 2-е изд. – М.: ИД «Вильямс», 2007. – 1408 с.
32. <http://www.osp.ru/os/2000/12/178361/> – эмпирическое сравнение семи языков программирования (дата обращения 25.06.2013).
33. http://en.wikipedia.org/wiki/Динамическое_программирование_язык – динамические языки программирования (дата обращения 24.06.2013).
34. <http://en.wikipedia.org/wiki/YAML> – справка по YAML (дата обращения 24.06.2013).
35. <http://www.yaml.org> – сайт спецификаций языка разметки YAML (дата обращения 24.06.2013).
36. Рыбина Г.В., Паронджанов С.С. Моделирование процессов взаимодействия интеллектуальных агентов в многоагентных системах // Искусственный интеллект и принятие решений. – 2012. – № 1. – С. 3–15.