

УДК 681.324

Н.Н. Глибовец, А.Н. Глибовец, А.Н. Корень, Ю.В. Рощенко

## Принципы реализации учебных взаимодействий в системах дистанционного образования

Описана концепция построения программной системы поддержки электронного обучения на основе агентной технологии, разработанная в рамках проекта создания программных средств поддержки колаборативного обучения в Киево-Могилянській академії. В основу системи покладено принцип навчальної взаємодії.

A concept of constructing a programming system of the E-Learning support on the basis of the agent technology is described. It is developed in the framework of the project of creating a software for the support of a collaborated learning in the Kyiv-Mohyla Academy. A principle of the learning interaction is the basis of the system.

Описано концепцію побудови програмної системи підтримки електронного навчання на базі агентної технології, розроблену в межах проекту створення програмних засобів підтримки колаборативного навчання в Києво-Могилянській академії. В основу системи покладено принцип навчальної взаємодії.

**Введение.** Опыт создания современных программных систем поддержки дистанционного образования (ПСПДО) свидетельствует, что такую систему необходимо рассматривать только как интеллектуальную сеть или некоторую онтологическую систему, поддерживающую интерактивные взаимодействия [1]. Согласно [2] ПСПДО удовлетворяет таким основным требованиям: настраиваться на эффективное выполнение запросов пользователей; оптимально выполнять основные операции с базами данных и знаний по сохранению, систематизации и передаче информации; архитектура услуг должна обеспечивать работу с разными программными платформами и ориентироваться на качественную передачу значительного потока информации мультимедийного типа. С позиций функциональности, учитывая возможность использования в ПСПДО агентных технологий [3], для описания взаимодействия подсистем характерно использование ролевого принципа задания предметной области знаний системы и создания систематизированного представления об этом.

Архитектурно ПСПДО – интегрированная среда, построенная с использованием современных возможностей Web-технологий, обеспечения как удаленной учебы, так и «традици-

онной», названной колаборативным обучением (*blanded learning*), с развитыми средствами тестирования знаний и обеспечения коммуникативного интерактивного взаимодействия между базовыми компонентами системы (слушатель, преподаватель, администратор, агент).

Для формирования программной платформы поддержки украинской системы электронного образования и достойной ее интеграции в мировую систему открытого обучения необходима развитая единая образовательно-учебная среда с использованием принципа унификации решений на базе международных стандартов. Следует объединить усилия отдельных университетов Украины по созданию программных систем поддержки непрерывного обучения. О реальности создания такой ПСПДО, учитывающей национальные особенности образования, свидетельствуют разработка «Академик» Львовского банковского института [4], распределенная система обучения НУ «Львовская политехника» [5], системы Национального технического университета «Киевский политехнический институт», Национального университета «Киево-Могилянская академия» и Киевского национального университета имени Тараса Шевченко, находящиеся на стадии завершения.

Большинство этих ПСПДО, к сожалению, почти не рассматриваются комплексно. Поэтому достаточно естественна проблема создания модели ПСПДО, обеспечивающей целостный взгляд на дистанционное (электронное) обучение, образцы которого отражает международный опыт. Для разработки спецификаций и рекомендаций по расширению использования коммуникационных и информационных технологий в образовании создан комитет стандартизации *IEEE LTSC (P1484 – Learning Technology Standart Committee)* разработки архитектуры и технологий образовательных систем (*Learning Technology Systems Architecture – LTSA*) [6] и международный образовательный консорциум *IMS Global Learning Consortium* [7].

В соответствии с идеологией *LTSA* для разработки технологических систем дистанционного образования целесообразны такие функциональные составляющие:

- разработка электронных вариантов учебных курсов и пособий с развитыми средствами самоконтроля усвоения материала и возможностями интерактивного взаимодействия между основными участниками учебного процесса (слушатель, преподаватель, администратор);
- управление учебным контентом, включая его доставку;
- обеспечение простого интерфейса получения знаний из базы знаний поддержки учебного процесса средствами Интернета;
- администрирование процесса обучения;
- индивидуальное планирование обучения, контроль и оценивание индивидуальных знаний слушателя;
- контроль и оценивание качества преподавания; коммуникация поддержки обучения и администрирования.

Для построения функциональной модели образовательной обучающей системы конкретного учебного заведения уточним *пять основных уровней архитектуры LTSA*.

На *первом* уровне выделяют два элемента – *объект* и *среда* обучения. Рассматривается влияние среды на обучаемого в процессе их взаимодействия при передаче знаний разными сред-

ствами. На втором уровне определяются особенности разработки, связанные с участием человека. Формулируются задачи разработки интерфейса взаимодействия среды обучения со слушателем, определения языка общения. На *третьем* уровне уточняются особенности использования системных компонентов *IEEE 1484.1*, выделяют базовые процессы (объекты), таблицы данных (структуры данных) и средства их хранения, а также основные информационные потоки. На *четвертом* уровне зафиксированы модели организации обучения. Выделяются подсистемы и устанавливается информационная нагрузка между ними с ориентацией на особенности выбранной модели и технологии обучения. На *последнем* уровне определяются основные требования, формируются базовые функции, строится концептуальная модель и определяется семантика системы обучения. Для образовательной технологической системы ввиду детального описания основных элементов обеспечивается интероперабельность кодов, форматов данных, протоколов, уровней взаимодействия и интерфейсов прикладного программирования (*APE*).

Модель организации образовательной среды полностью определяет алгоритм реализации процессов взаимодействия, состоящий из таких шагов.

1. Ученик выбирает стиль, тактику и стратегию, методы обучения согласно предложениям и возможностям среды обучения.

2. Проводится начальное тестирование знаний обучаемого и устанавливаются его психологические особенности с учетом предыстории и цели обучения. Полученная информация заносится в базу данных.

3. Преподаватель подбирает учебные ресурсы в базе знаний и устанавливает режим их доставки ученику, включая поставку программного инструментария поддержки интерфейса со средой обучения.

4. Реализация предоставления образовательных ресурсов путем использования интерактивных методов мультимедиа.

5. Организация промежуточного и конечного тестирования для оценивания полученных

знаний слушателем для накопления информации в базе данных об успеваемости слушателя.

6. Оценка уровня сертификата подтверждения полученных знаний слушателем в системе учебы.

Согласно [8] универсальность *L TSA* обеспечивает ее использование в качестве эталонной модели для разработки образовательных технологических систем произвольного типа.

### **Интеллектуализация процесса обучения**

Базис построения ПСПДО обеспечивает эффективное взаимодействие между учеником и средой обучения. Эта задача решается путем интенсификации интеллектуализации процесса обучения. В работе [9] выделены три модели обучения, которые можно успешно использовать.

*Когнитивная модель Нормана* [10], в основе которой – задание знаний посредством семантической сети [11]. Получение нового знания слушателем происходит путем добавления новой информации в уже существующую его семантическую сеть знаний. Успех усвоения знаний в таком процессе полностью зависит от количества взаимодействий с обучающим материалом и средой обучения.

*Модель гибкого обучения* [12]. В этом подходе процесс обучения превращается в исследовательскую деятельность, основная роль в которой отводится консультации с преподавателем по мере необходимости.

*Гипертекстовая модель* [13]. В основу модели положено понятие ассоциативности мышления. Заданная в явном гипертекстовом виде семантическая сеть знаний преподавателя дает студенту прямой доступ к нужному материалу с помощью броузинга.

В основу создания большинства интеллектуальных систем обучения положена алгоритмо-эвристическая теория обучения [14], развитая в работах [15, 16]. В этой теории считается, что любую познавательную деятельность можно проанализировать в терминах операций алгоритмической, полуалгоритмической, эвристической, полуэвристической природы. Существенный момент при этом подходе – построение

адекватной модели слушателя. Знание о том, каким желательно видеть ученика по завершении обучения, называют нормативной моделью обучаемого, или стандартом учебы, которую согласно [16] можно разбить на пять компонентов: предметный, функциональный, семантический, процедурный, операционный.

Часть нормативной модели, определяющей предметные знания, т.е. знание учебных курсов, называют *предметной моделью ученика*. В инженерии знаний такие знания называют экспертными. Для оптимизации процесса, формирующего способ достижения цели обучения, желательна структуризация знаний предметной модели в виде метазнаний. Здесь важно также определение «ролевых» функций знаний, т.е. функциональная структуризация путем перечня функциональных рубрик. Проведя структуризацию и разделив знания на *декларативные* и *процедурные*, получают *функциональную предметную модель* обучаемого. Декларативные знания описывают утверждения (факты) о свойствах объектов предметной области изучения и отношений между ними, т.е. они определяют содержательную часть предметных знаний, порождая *семантическую предметную модель* обучаемого. *Семантический факт* – наиболее мелкая единица знаний предметной области. *Процедурные знания* описывают порядок и характер превращений объектов предметной области и выполняют роль схем ориентировочных основ деятельности слушателя в процессе решения задач. Знания, с помощью которых формируется умение найти решение задачи, т.е. выработать способ действий для решения определенной задачи в процессе обучения, называют *операционными знаниями*.

Можно считать, что интеллектуализация учебного процесса ПСПДО заключается в выработке эффективных методов задания знаний предметной области как обучаемого, так и среды обучения путем их разделения на тематические, функциональные, семантические, процедурные, операционные и создание на основе этого авторизованных гипермедийных распределенных баз знаний.

## Подсистема получения знаний

Цель подсистемы – эффективный перенос знаний из базы знаний к обучаемому в удобной форме для усвоения материала, для чего необходимы такие действия.

- Обеспечить достаточный уровень построения и сохранения метазнаний. При описании метазнаний обязательно включаются требования как к учебному процессу, так и к процессу преподавания знаний.

- Трансформировать знания, переданные на вход системы в гипермедийный вид, удобный для сохранения и обработки как системой, так и пользователем.

- Преподавать знания в структурированном, удобном для восприятия виде. В структуризации учебных материалов целесообразны три фактора – *методологический, психологический, эргономический*.

Детальное описание системы передачи знаний не является целью этой работы, поэтому приведем схему (рис. 1), иллюстрирующую передачу знаний – введение знаний в систему, трансформацию неструктурированных знаний в учебные материалы и подачу их пользователям.

С помощью *группы подготовительного курса* (может состоять из преподавателя и методиста) знания преобразуются в структурированные учебные материалы, сохраняемые в системе и предоставляемые по требованию пользователя. Группа подготовки курса вводит сценарий определенного уровня учебного процесса, например, сценарий курса.

Система дает ученику возможность овладеть учебным материалом и следит за его работой, предоставляет пользователям возможность взаимодействовать между собой и объектами системы (например, ученик может обратиться за консультацией к экспертной системе).

Учебный процесс разделен на учебные материалы и сценарии. Этот подход – основной при создании системы. Для учебных материалов задано соотношение порядка (структура за-

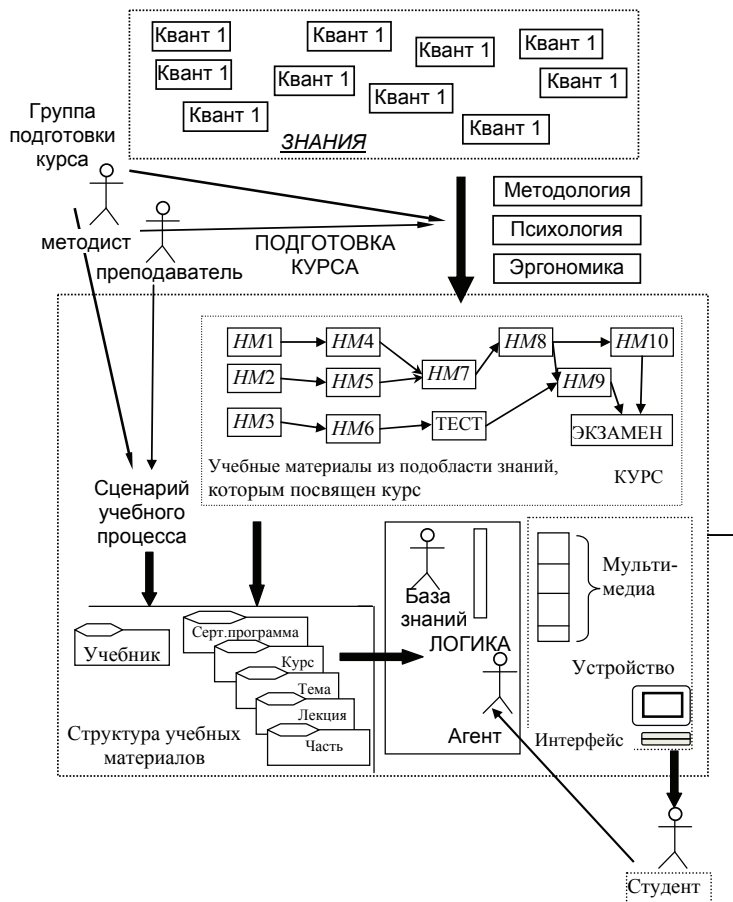


Рис. 1. Схема передачи знаний

висимостей) и метрики. Так, структура зависимостей может определять последовательность изучения материалов в рамках курса и последовательность прохождения курсов в сертификационной программе. Как метрики можно использовать максимальные оценки в баллах, время изучения материалов, сложность материалов и др. Так, на рис. 2 показана возможная структура овладения учебным курсом.

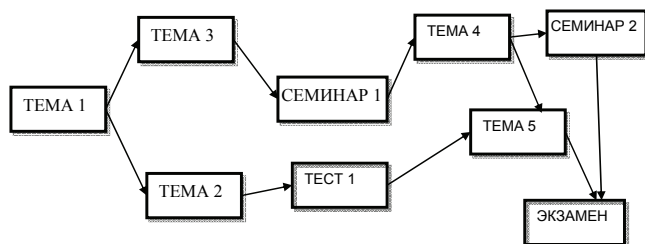


Рис. 2. Пример структуры учебных материалов курса

Такой подход дает возможность автоматизировать систему выбора овладением материалом курса (самый быстрый способ сдать курс, способ, который принесет как можно больше

знаний, прохождение курса с наилучшей оценкой), используя известные алгоритмы на графах.

Построение учебного процесса преобразуется в сценарий курса, определяющий последствие для обучаемого в возможных вариантах его поведения. Сценарий может быть разного уровня сложности. Простейший реализуется набором продуктивных правил типа *if-then-else*. Сценарий с расширенной семантикой включает в себя итерационные возможности, логические операции, действия над объектами и др. Структура и сценарий описаны с помощью сети Петри, тогда на учебные материалы с заданной структурой накладывается еще и семантика состояний, фишек и переходов. Эта сеть определяет возможные варианты прохождения учебного процесса. Так можно смоделировать в принципе недетерминированное, но с ограничением, поведение слушателя в системе дистанционного образования.

*Заданные интерфейсы* можно сгруппировать по следующим параметрам: гибкость, настроенность, удобство, доступность (для устройств и людей), универсальность, информативность, техничность. Уточним смысл этих параметров. *Информативность* – количество информации, способной пройти через интерфейс, по каждому направлению – своя информация. *Гибкость* определяется способностью к видоизменению по желанию пользователя; количеством настроек, изменяющих вид и поведение элементов интерфейса. *Удобство* – число, значение которого обратно пропорционально количеству действий, необходимых для доступа к возможностям системы. *Эргономичность* – прямо пропорциональна максимальному эффективному времени, которое пользователь может провести с системой. Сохранение определенного уровня информативности для всех поддерживаемых устройств характеризует доступность. *Универсальность* включает в себя и возможности использования интерфейса людьми с физическими изъянами.

С учетом этих характеристик избран *XML* как язык передачи промежуточных данных и сохранения изначальных сущностей (учебник, тест).

## Проектирование ПСПДО

Развитие новых информационных технологий позволяет постепенно устранить основной тормоз развития непрерывного дистанционного образования – формирование адаптивной структуры обучения в зависимости от индивидуальных особенностей ученика путем адекватной поддержки развития методик интерактивного общения учителя и ученика. ПСПДО должны обеспечивать: единство методического, организационного, информационного, программного, технического уровня сред; систематизацию инфоресурсов, создание больших распределенных баз знаний; построение стандартов (норм), которые облегчили бы поиск, обмен и разнесение учебных материалов с возможностью интерактивного общения. Известно [17, 18], что к таким технологиям можно отнести технологию многоагентных систем.

Система, в которой несколько агентов могут общаться между собой, обмениваться текущей информацией, взаимодействовать, называется *мультиагентной*. Такие системы относят к распределенным системам искусственного интеллекта [19]. В них для решения задачи используется несколько интеллектуальных агентов, т.е. задача разбивается на подзадачи, решение которых распределяется между отдельными агентами. В случае ПСПДО каждого из участников процесса обучения можно представить в виде агента [3].

Известно, что процесс проектирования и моделирования при создании программного обеспечения может занимать до 80 процентов всех ресурсов. Правильно спроектированная непротиворечивая модель выполняет критическую роль во всей разработке. Внутренне несогласованная или неполная модель может привести к полной неудаче проекта. К тому же, в последние годы появились развитые языки поддержки моделирования, в первую очередь – унифицированный язык моделирования *UML* [20].

Средства автоматической кодогенерации позволяют переводить модели на *UML* в исходный код объектно-ориентированных языков программирования, что еще более ускоряет процесс разработки. В настоящее время мно-

гие CASE-средства автоматизируют процесс анализа и проектирования в UML, например, Rational Rose [21].

Поэтому к процессу создания ПСПДО можно подойти таким образом: создать экспертную систему с использованием распределенной базы знаний, в которой построение выводов возложено на некоторую мультиагентную систему. Для описания проектирования такой системы используем Rational Rose Professional Edition. Мультиагентную систему можно создать, пользуясь программной платформой Java Agent Development Environment (JADE), преимущество которой – в ее платформенной независимости и возможности использования удаленного GUI-интерфейса.

Для создания функциональной модели мультиагентной системы желательно отследить все возможные взаимодействия в системе, физическая структура которой приведена на рис. 3.

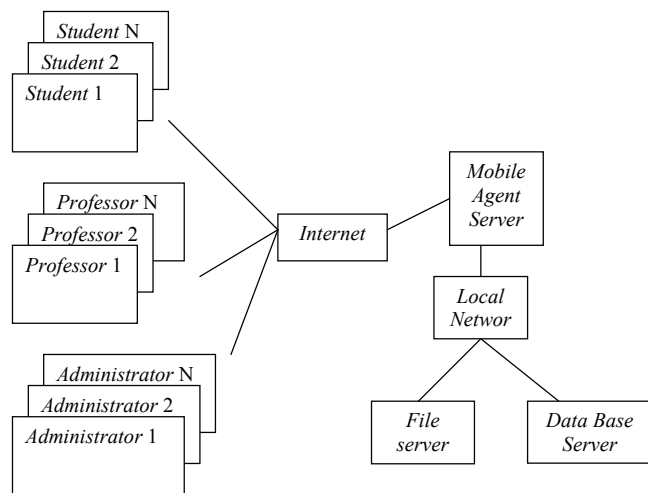


Рис. 3. Физическая структура системы

Основу таких взаимодействий составляет учебное взаимодействие.

### Учебные взаимодействия

Под учебным взаимодействием будем понимать любые действия ученика или среды, следствием которых является получение знаний и навыков учеником. Можно выделить такие виды учебного взаимодействия в системах дистанционного образования:

- получение знаний путем пересмотра учебных материалов из гипермедийной базы знаний;

- стимуляция навыков (семинары, практические занятия, лабораторные работы);
- общение с преподавателями, другими слушателями, системой;
- оценивание.

Понятие учебного взаимодействия должно обобщать эти виды взаимодействий, реализованные разными способами, в частности такими, как броузинг, видеоконференция, чат, форум, электронная почта и другие с учетом разных психологических аспектов. Взаимодействие также должно быть открытым к расширению, т.е. должно давать возможность адекватно описывать другие типы взаимодействий, еще разрабатываемые или реализованные вне системы. Примером такого взаимодействия может служить «внеклассное» общение слушателя и преподавателя, которое в реальной жизни никак не протоколируется и не предусмотрено программой, но влияет на результат.

Три последних вида взаимодействий можно обобщить с помощью абстракции задания. Уточним эти понятия.

Под взаимодействием (*Interaction*) будем понимать следующий набор: тип, участники, коммутатор:

$$Interaction = \{Type, Actors, Commuter\}.$$

Тип (*type*) – набор атрибутов взаимодействия относительно существенных параметров. Начальный тип может состоять из одного поля *TimeSensitivity* (временной чувствительности) – максимально допустимое время задержки реакции объекта на сообщение.

Участник (*actor*), как участник учебного процесса, имеет определенный набор атрибутов и функций, а также владеет определенными правами доступа к системным функциям. Он определяется набором совместимых интерфейсов и временной чувствительностью:

$$Actor = \{Interface1, \dots, INTERFACEN, TimeSensitivity\}.$$

Коммутатор (*Commuter*) – объект, отвечающий за прохождение и превращение информационных потоков. Коммутирует информационные потоки и определяется набором интерфейсов, с которыми он совместим, средой и временами задержки:

$Commuter = \{Control, Interfaces, Environment, Type\}$ .

Под управлением (*Control*) понимается модель коммутации информационных потоков. Среда (*Environment*) характеризуют надежность передачи, наличие или отсутствие предыстории, возможности передачи разных типов медиа:

$Environment = \{Reliability, History, MediaSet\}$ .

Интерфейс определяется как набор возможностей прохождения медиа в обе стороны:

$Интерфейс = \{InCapability, OutCapability\}$ ,

где  $InCapability = \{MediaSet, TimeSensitivity\}$ ,

$OutCapability = \{MediaSet, TimeSensitivity\}$ ,

$MediaSet = \{Text, Sound, Video, Movement, Image, Model\}$

и *Text* помечает текст, *Image* – изображение, *Sound* – звук, *Video* – видео,

*Movement* – движение, *Model* – интерактивная модель.

В рамках данного подхода разные типы медиа специально не приводятся к гипертексту, потому что в реальных средах не всех их можно передать. Схема взаимодействия показана на рис. 4.

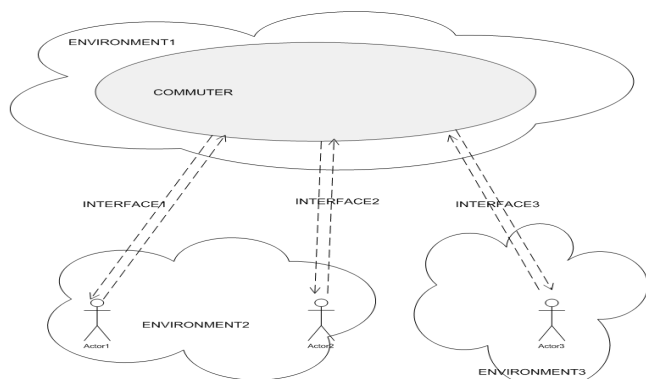


Рис. 4. Схема взаимодействия

Основное учебное взаимодействие – задание как коммуникативное взаимодействие – проводится для оценивания уровня знаний и умений слушателя. Задание наследует свойства взаимодействия и обобщает такие понятия, как семинар, практическая работа, тест, экзамен. Задание описывают следующим образом (свободная нотация):

## ЗАДАНИЕ

Цель работы

Разделы учебного материала, проверяемого или закрепляемого

Необходимые для работы объекты

Содержание работы

Результат по схеме 100%, баллов

Дополнительные сведения (ссылка на учебные материалы, необходимые для выполнения задания, методические указания)

ЗАДАЧИ

ЗАДАЧА\_1

ТИП

ПОСТАНОВЩИК

ОЦЕНЩИК

Результат на 100%, баллов

Обратная связь

## Тестирование как реализация взаимодействия «задания»

Одна из главных подсистем любой системы образования контролирует уровень усвоения знаний и названа подсистемой тестирования. Специальный комитет Американской ассоциации оценивания разработал пять общих принципов оценивания знаний: систематичность исследования, компетентность, полноценность/честность, уважение к людям, ответственность за общее и общественное благосостояние [12].

**Жизненный цикл тестовых вопросов.** Произвольные вопросы в системе *Web*-тестирования имеют свой жизненный цикл (таблица), который состоит из трех этапов: подготовка (начало активной жизни), передача (середина активной жизни), оценивание (конец активной жизни). Каждый из этих этапов обеспечивается определенным набором Интернет-технологий.

Жизненный цикл вопросов в системе *Web*-тестирования

Начало	Середина	Конец
<i>Подготовка:</i>	<i>Передача:</i>	<i>Оценивание:</i>
Авторская разработка вопросов	Представление	Оценка
Сохранение вопросов в системе	Взаимодействие	Выставление баллов
Выбор вопросов для представления как часть теста	Получение ответа	Обеспечение обратной связи

Начало активной жизни вопроса приходится на стадию авторской разработки. Задача системы на этой стадии состоит в обеспечении

автора-разработчика тестов соответствующими технологиями и средствами, необходимыми для создания тестов.

Следующая стадия – сохранение всех разработанных вопросов (содержание и метаданные) в системе. Активная жизнь сохраненного вопроса начинается тогда, когда его выбирают для представления как часть текста. Этот выбор может выполняться статично (преподавателем во время разработки курса) или динамично (системой в реальном времени).

Этап передачи вопроса выполняется так: система представляет вопрос, обеспечивает интерфейс для слушателя, намеренного отвечать на вопрос, и получает на него ответ.

На этапе оценивания система должна выполнять такие задания: оценить вопрос; выставить необходимое количество баллов; сообщить о результатах тестирования.

#### *Этап подготовки*

Начало активной жизни вопроса приходится на стадию авторской разработки. Задание системы *Web*-тестирования на этой стадии заключается в обеспечении автора (разработчика тестовых вопросов) соответствующими технологиями и средствами, необходимыми для создания тестов. Следующая стадия – сохранение всех разработанных вопросов (содержание и метаданные) в системе. Активная жизнь сохраненного вопроса начинается, когда его выбирают для представления как часть текста. Этот выбор можно выполнять статически (преподаватель во время разработки курса) или динамически (система в реальном времени). Вопросы в системе создают авторы. Каждый вопрос должен иметь следующие атрибуты: собственно вопрос, варианты ответа, индикация правильного ответа. В дополнение к этому автор может добавить специальные метаданные (связанные темы, ключевые слова, тексты, значимость или сложность вопроса, допустимое время, число попыток и др.).

Различают два способа хранения вопросов: формат представления и внутренний формат. Хранение вопроса в формате представления означает хранение его в форме куска *HTML*-кода (обычно *HTML*-формы). Это так называе-

мые статические вопросы. Хранение вопроса во внутренней форме – значит хранение его во внутренней базе данных, где разные части вопроса (именно вопрос, ответы и реакции) сохраняются в разных полях записи. Вопрос, который видит студент, генерируется во время доставки. Такая форма сохранения позволяет представлять вопрос более гибко, т.е. один и тот же вопрос можно представить, например, в виде вопроса с полями для заполнения или для поливариантного выбора, либо с разными интерфейсами. Лучший вариант сохранения вопросов – хранение их в *XML*-файлах, что позволяет упростить структуризацию вопроса, ускорить доступ и обработку данных.

*Этап передачи вопроса.* Наиболее развитой технологией при передаче вопросов является комбинация *HTM*-форм и *CGI* испытательных скриптов. Основные достоинства такой технологии состоят в том, что ученик получает лишь готовую *HTML*-страницу и не видит весь процесс, проходящий на стороне сервера: формирование вопроса и получение, оценивание и запись результатов теста. Но эта технология не срабатывает, когда связь с пользователем очень замедлена или не обрывается вообще. Более новой технологией доставки и проверки вопросов являются *Java*-скрипты. Основной плюс таких технологий – то, что они могут работать *off-line*. Минус – доступ к коду программы (можно сломать программы и изменить результаты теста). Потому при создании системы нужно выбирать между этими двумя типами. В большинстве случаев преимущество отдается первому типу вследствие его простоты.

*Этап оценивания.* На этом этапе система должна выполнить такие задания: оценить вопрос как правильный (либо неправильный), частично правильный, выставить необходимое количество баллов, полученное студентом при ответе на вопросы; уведомить студента о результатах тестирования. Обычно правильные (или неправильные) ответы предлагаются во время авторизации, а значит, проверка либо жестко привязана к вопросу, либо проводится простым сравнением. Обратная связь в системе дистанционного обучения очень важна, поскольку уче-



ник может увидеть допущенные во время теста ошибки. Этот этап должен отличаться в зависимости от этапа обучения, когда он был применен. Так, если этот тест нужен для самооценки после какой-то лекции, либо это – текущий тест, то он должен иметь обратную связь, т.е. сопровождаться неправильными ответами и нотацией учителя ко всему тесту. Например, материал усвоен плохо, рекомендуется еще раз повторить лекцию № 5 и обратить особое внимание на определенные вопросы. Тогда студент знает, что обязан выучить, и его знания станут более полными. А финальный тест обязан содержать лишь оценку. Результаты теста должны быть автоматически записаны в соответствующую форму. Система должна предоставлять правильно распределенный доступ к оценкам для студентов, преподавателей и администраторов. Обычно ограничения накладываются согласно университетским правилам.

**Заключение.** Приведенная концепция реализации учебных взаимодействий в программных системах поддержки электронного образования позволяет выделить минимум основных, существенных для системы взаимодействий, и значительно упростить проектирование и разработку системы.

Анализ модели взаимодействий привел к следующим выводам по построению национальной платформы ПСПДО.

- Для эффективного взаимодействия с мировым сообществом необходимо соответствие международным стандартам по архитектуре и формату представления данных в депозитариях сохранения учебных материалов.

- Учебный процесс должен быть разделен на *учебные материалы* и *сценарий*. Это – основное условие при создании системы.

- Необходимо обеспечить распределенный доступ участников процесса к возможностям системы. Поэтому ПСПДО необходима для работы распределенная платформа. Отсюда к проблемам разработки присоединяется проблематика распределенных систем.

- Поскольку учебный процесс переносится в новую среду, достаточно актуальной может

оказаться проблема методологического контроля этого процесса. Готовых методологических наработок недостаточно для современного применения. Целесообразно пересмотреть методологию в аспекте новых применений.

- Использование компьютерных технологий может иметь разные последствия. К негативным можно отнести в частности стереотип участников относительно учебного процесса (обучение на компьютере может не восприниматься даже подсознательно), возможность «нечестного» поведения с системой (слом, подмена ученика). Поэтому при построении ПСПДО необходимо сознательно усиливать использование позитивных факторов и преодолевать негативные.

- ПСПДО должна удовлетворять определенным эргономическим требованиям, чтобы участники процесса имели возможность работать с системой столько времени, сколько потребуется для их обучения, без усталости и вреда для здоровья. Каждый из участников должен иметь возможность подобрать себе удобный интерфейс. Кстати, в значительной степени к эргономическим аспектам относят требования к аппаратной части.

- Как показывает практика, в мире программного обеспечения большой успех имеют разработки, использующие семантическую связь с уже известными пользователю сущностями, т.е. имеющими системную метафору из реального мира. Яркий пример – Виртуальный университет.

- Система должна не только хранить и предоставлять информацию, подготовленную преподавателями, но и проявлять себя активным участником учебного процесса: с системой можно проконсультироваться, предполагая адекватную реакцию и адаптивное ее поведение (в частности во время тестирования). Система должна быть способна к самосовершенствованию, выявлению попыток обмана, стойкостью к ошибкам и разрушению. Для разработчиков это значит, что они должны максимально гибко реализовать все функции, используя, по возможности, интеллектуальные или эвристические алгоритмы.

1. Patel A., Kinshuk & Russel D., (2000) Intelligent Tutoring Tools for Cognitive Skill Acquisition in Life Long learning, Educational Technology & Society, Ed. by Sinitsa K. // J. of IFETS and IEEE-LTTF, Available – <http://ifets.ieee.org/peridical/>
2. Chen M. A methodology for characterizing computer-based learning environments // Instructional Science. – 1995. – N 23. – P. 183–220.
3. Глибовець М.М. Використання агентних технологій в системах дистанційної освіти // УСИМ. – 2002. – № 6. – С. 69–76.
4. Чаплига В.М., Абашина Н.М., Вільдштейн Д.В. Система дистанційного навчання (СДН) «Академік» // 36. матеріалів міжнар. наради «Телематика і безперервна освіта», Київ, 2001. – С. 174–176.
5. Проектування Web-орієнтованих розподілених навчальних систем / Ю. Рашкевич, Д. Пелешко, М. Пасека, А. Стецюк // Там же. – С. 143–152.
6. IEEE LTSC. – <http://ltsc.ieee.org>
7. IMS. – <http://www.imsproject.org>
8. Андреев А.А. Введение в дистанционное обучение. – <http://www.iet.mesi.ru/broshur/broshur.htm>
9. Грицай В.П. Информационная технология «Динамический гипертекст ТЕТ-А-ТЕТ» // Вестн. Междунар. Соломонова ун-та, 2000. – № 2. – С. 24–32.
10. Norman D.A., Defending Human Attributes in the Age of the Mashine. – New York: Addison Wesley Publ. Voyager, 1995. – 304 p.
11. Глибовець М.М., Олецький О.В. Штучний інтелект. – К.: КМ Академія, 2002. – 365 с.
12. Nunan T. Flexible delivery – a discussion of issues. University of South Australia: Distance Education Centre, 1995. – <http://www.londonmet.ac.uk/.../flexible.../nunan.cfm>
13. Эпштейн В.Л. Введение в гипертекст и гипертекстовые системы. – <http://www.ipu.rssi.ru/publ/epstu.htm>
14. Landa L. Algorithmization in Learning and Instruction, Englewood Cliffs, NJ: Educational Technology Publ., 1974. – 713 p.
15. Атанов Г.А. Деятельностный подход в обучении. – Донецк: «ЕАИ-пресс», 2001. – 160 с.
16. Атанов Г.А. Пятикомпонентная предметная модель обучаемого // Educational Technology & Society. – 2001. – N 4 (1). – P. 111–124.
17. Гороховський С.С. Агентні технології: Спроба критичного огляду // Наук. записки НаУКМА. Комп'ютерні науки. – 2000. – Т. 18, Ч. 2. – С. 34–45.
18. Городецкий В.И., Грушинский М.С., Хабалов А.В. Многоагентные системы (Обзор) // Новости искусственного интеллекта. – 1998. – № 2. – С. 64–116.
19. Смирнов А.В., Шереметов Л.Б. Многоагентная технология проектирования сложных систем // Автоматизация проектирования. – 1998. – Ч. 1: 3(9). – С. 45–50; 1999. – Ч. 2: 1(11). – С. 42–46.
20. Крэг Ларман. Применение UML и шаблонов проектирования. Введение в объектно-ориентированный анализ и проектирование. – М.: Вильямс, 2002. – 624 с.
21. Терри Кватрани. Rational Rose 2000 и UML. Визуальное моделирование / Visual Modeling with Rational Rose 2000 and UML – М.: ДМК, 2001. – С. 12–24

Поступила 13.04.2010

Тел. для справок: (044) 416-4515, 463-6985 (Київ)

E-mail: [glib@ukma.kiev.ua](mailto:glib@ukma.kiev.ua)

© Н.Н. Глибовец, А.Н. Глибовец, А.Н. Корень,

Ю.В. Рощенко, 2010