

УДК 519. 72

ИДЕНТИФИКАЦИЯ ПОЛЬЗОВАТЕЛЕЙ В СИСТЕМЕ ДИСТАНЦИОННОГО ОБРАЗОВАНИЯ НА ОСНОВЕ ПРОГРАММНЫХ АГЕНТОВ

В. А. Филатов¹, Р. В. Семенец²¹ ХНУРЭ, г. Харьков, Украина, Filatov_val@ukr.net² ХНУРЭ, г. Харьков, Украина, Ars_sem@ukr.net

В статье обосновано применение агентных технологий для решения задачи идентификации пользователей в системе дистанционного образования, построена математическая модель пользователя и предложен вариант ее реализации на основе агентного подхода, приведена логическая схема интеллектуального программного агента.

ДИСТАНЦИОННОЕ ОБРАЗОВАНИЕ, ТЕСТИРОВАНИЕ, СЛУШАТЕЛИ, ИДЕНТИФИКАЦИЯ, ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ, ПРОГРАММНЫЕ АГЕНТЫ, ФРЕЙМОВАЯ МОДЕЛЬ, МОДЕЛЬ ПОЛЬЗОВАТЕЛЯ.

Введение

Информационные технологии стали неотъемлемой частью существования современного общества. Сейчас трудно представить такую область человеческой деятельности, где не применялись бы персональные компьютеры с комплексом программных средств информационной поддержки. К одной из них — наиболее динамично развивающейся — относится сфера образования. За последние годы качественно изменились все составляющие технологии образования: от методики преподавания до технических средств обучения. Являясь следствием объективного процесса информатизации, интенсивно развивается принципиально новое направление в обучении — дистанционное образование.

Важно отметить, что это направление получило развитие не только в рамках национальных систем образования, но и в коммерческих компаниях с преимущественной ориентацией на подготовку в области бизнеса. Частные корпоративные образовательные сети созданы такими компаниями, как IBM, General Motors, Ford и др. Многие из этих образовательных систем намного опережают по сложности и объему системы, созданные в университетах.

1. Постановка проблемы

В технологии дистанционного образования используется следующая схема обучения и контроля знаний. Обучаемый получает комплекс методических материалов по теме: лекционный материал, набор практических заданий, лабораторный практикум и набор тестов как элемент контроля усвоения материала. Тесты с результатами возвращаются преподавателю для оценки степени усвоения материала.

Таким образом, в деканате дистанционного обучения на каждого студента формируется база данных с результатами тестовых задач по разным дисциплинам, представленная на рис. 1.

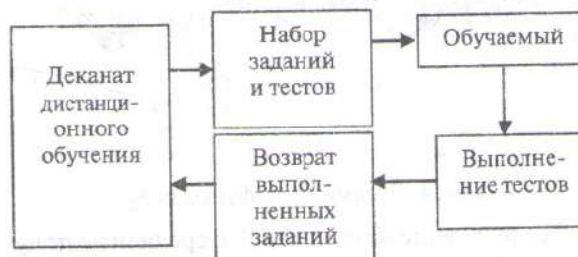


Рис. 1. Схема организации процесса обучения в системе дистанционного образования

Набор выполненных студентом тестовых задач может быть использован для получения индивидуального почерка — модели пользователя в системе дистанционного образования. Для этого тесты, которые выполняет слушатель, должны содержать служебный модуль, регистрирующий «манеру поведения» тестируемого. В качестве критерия построения модели пользователя может быть выбран такой параметр, как время обдумывания вопроса и формирования ответа. Наличие таких тестирующих программ позволяет проводить тестирование слушателей как в присутствии преподавателя, так и в автономном «домашнем» режиме. В первом случае выполнение теста можно рассматривать как режим «обучения», или настройки модели слушателя. Вторым вариантом — идентификация пользователя. Реализация такой задачи в системе дистанционного образования позволит преподавателю оценивать с определенной степенью достоверности авторство «домашних» тестов.

Рассмотренная выше технология обладает одним существенным недостатком. Тесты, выполняемые одним слушателем по разным предметам, разработаны разными преподавателями, структура и технология тестирования в которых, в общем случае, может быть совершенно разной. Поэтому критерий «время обдумывания одного вопроса» в системе идентификации слушателя будет неэф-

фективен. Сохранить привлекательность метода идентификации пользователя на основе временных показателей можно, применив модульный подход. Для этого разрабатывается одна универсальная оболочка, к которой может быть подключен любой тестирующий программный модуль. Однако, чтобы «войти» в специализированный тест по дисциплине, необходимо пройти «входной» контроль или «входное тестирование». Структура такой системы представлена на рис. 2.

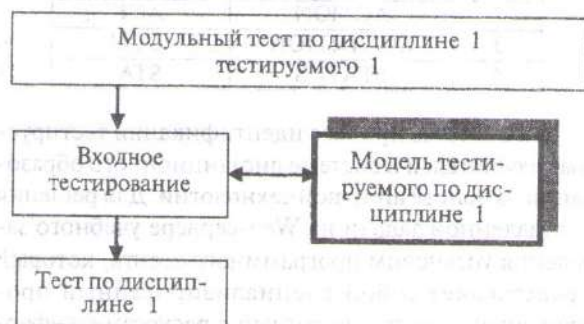


Рис. 2. Схема универсального модульного теста

Наличие модуля «Входное тестирование» обязывает всех слушателей проходить универсальный тест при выполнении любого теста по изучаемой дисциплине. Таким образом, модульный тест будет сохранять в себе результаты как теста по дисциплине, так и индивидуальный почерк на основе входного теста. Затем необходимо обработать все модульные тесты одного слушателя на предмет сравнения результатов входных тестов. На рис. 3 изображена схема анализа модульных тестов.



Рис. 3. Подсистема анализа модульных тестов

В рассмотренной технологии дистанционного образования в качестве универсального входного теста может быть предложен «тест на клавиатуре». Слушателю последовательно выводятся на экран группы слов, которые он должен правильно набрать на клавиатуре. В таком тесте эффективно может быть использован критерий «время выполнения за-

дания» и реализован метод идентификации на основе функции подобия моделей поведения тестируемых.

Цель исследования. Целью проводимых исследований является разработка математической модели идентификации пользователей в системе дистанционного образования и ее реализация на основе технологии программных агентов.

2. Разработка математической модели

Для разработки математической модели пользователей на первом этапе необходимо предусмотреть тестирование в присутствии преподавателя.

Пусть $U = \{u_k\}, k=1, K$ — множество тестов по дисциплинам, которые необходимо сдать студенту в присутствии преподавателя, $D = \{d_k\}, i=1, L$ — множество заданий во входном тесте.

Множество D является полным множеством всех заданий входного теста. Отношение слушателя, выполняющего тестирование по первой дисциплине, к полному множеству заданий входного теста D представляется в виде бинарного вектора, содержащего единичные элементы в той позиции, которой соответствует ответ в отведенный регламент времени на это задание, и нулевые в противном случае. Интервалы времени переходов из 1 в 0 по каждому заданию теста определяются разработчиком теста. В результате тестирования формируется бинарная матрица $B = \|b_{kj}\|$, проиндексированная по двум множествам: тестируемых дисциплин $U = \{u_k\}$ и полного множества заданий во входном тесте $D = \{d_k\}$.

Далее на основании анализа матриц $B_l = \|b_{kj}\|, l=1, L$, где L — количество студентов, одним из методов кластерного анализа производится настройка моделей слушателей в системе дистанционного образования.

Студенты могут выполнять тесты по изучаемым дисциплинам в домашних условиях без преподавателя. Присылаемые результаты тестов по изучаемым дисциплинам будут содержать в себе служебную информацию — данные входных тестов.

Следующий этап — сравнение данных входного теста с моделью слушателя для оценки степени самостоятельности выполнения заданий.

Для получения количественной характеристики степени общности модели 1-го слушателя с данными очередного входного теста воспользуемся понятием меры подобия $0 \leq S_i \leq 1$. Меру подобия предлагается вычислять при помощи выражения:

$$S_i = (p_1) / (p_1 + p_{10} + p_{01}), \quad (1)$$

где p_1 — количество общих элементов модели и входного теста; p_{10} — количество элементов, принадлежащих модели, но отсутствующих во входном тесте;

p_{01} — количество элементов, отсутствующих в модели, но принадлежащих модели входного теста.

Одним из способов программной реализации рассмотренной выше математической модели идентификации пользователей в системе дистанционного образования может быть подход на основании технологии программных агентов.

3. Обоснование модели мультиагентной системы

Агентные технологии имеют большое количество преимуществ по сравнению с традиционными технологиями для решения специфических проблем. Адаптированные в наиболее общем виде к объектно-ориентированным технологиям, включая и технологию распределенных объектов, они представляют собой мощное и эффективное инструментальное средство для решения специфических задач информационной поддержки. Перспективным направлением является интеграция агентных технологий с традиционными информационными технологиями. Такие свойства агентных систем позволяют расширить область их применения и на сферу дистанционного образования [1].

Мультиагентной системой назовем многокомпонентную систему, состоящую из программных агентов, выполняющих локальные задачи информационной поддержки для достижения общей цели, — управление распределенной информацией.

Основные составляющие мультиагентной системы:

Агент — приложение, функционирующее на локальном компьютере и используемое в общем случае для реализации доступа и администрирования информационных ресурсов.

Система управления агентами (менеджер агентов) — программа-администратор мультиагентного пространства, осуществляющая следующие функции:

- контроль работы системы;
- дистанционное управление поведением мультиагентной системы;
- модификация и настройка функций программных агентов на удаленных компьютерах;
- накопление и анализ информации, полученной от агентов в процессе их функционирования.

Конструктор агентов — подсистема, обеспечивающая логическое и физическое проектирование агентов, их генерацию и модификацию [2, 3].

Модель интеллектуального агента построена на основе фрейма. В общем случае модель фрейма имеет вид:

$$FR = [(R_1, A_1), (R_2, A_2), \dots, (R_n, A_n)], \quad (2)$$

где FR — имя фрейма; R — имя слота; A — значение слота.

На основании модели фрейма (2) может быть представлена концептуальная модель программного агента в терминах <объекты>, <условия>, <действия>, <приоритет>. Каждый слот будет формироваться из четырех атрибутов базовых сущностей, представленных в табл. 1.

Таблица 1

Типы слотов

№	Сущности	Имя
1	OBJECT	OBG
2	ACTION	ACT
3	CONDITION	CON
4	STATUS	STA

Рассмотрим процесс идентификации тестируемых слушателей в системе дистанционного образования на основе агентной технологии. Для решения поставленной задачи на Web-сервере учебного заведения установим программного агента, который представляет собой специализированный программный модуль, связанный с ресурсами информационной системы, — базой данных зарегистрированных слушателей. Его первой задачей является проверка соответствия вводимого логина и пароля записям базы данных. Второй задачей программного агента является осуществление входного тестирования слушателя и программная реализация математической модели на основе анализа клавиатурного почерка. При этом один программный агент может решать несколько задач по управлению доступом к ресурсам данного персонального компьютера.

4. Языковые средства создания программных агентов

Основными операторами языка создания программных агентов являются следующие операторы:

CREATE AGENT (DROP AGENT)
CREATE SLOT (DROP SLOT)

Эти операторы используются для создания, изменения и уничтожения агентно-ориентированных структур [4].

Создание программного агента.

CREATE AGENT [name | AUTHOR creator-identifier]

Таким образом, если проектировщиком схемы под именем test будет пользователь Admin, то данный оператор будет выглядеть следующим образом:

CREATE AGENT test AUTHOR Admin

Схема удаляется с помощью оператора DROP AGENT, который имеет следующий формат:

DROP AGENT [RESTR | CASCAD]

Если указано ключевое слово RESTR (именно оно принимается по умолчанию), схема взаимосвязей программных агентов должна быть пустой, ина-

че выполнение операции будет отменено. Ключевое слово CASCAD определяет автоматическое удаление всех связанных с данным агентом других программных агентов.

После создания структуры программного агента можно приступить к созданию функциональных слотов-отношений, входящих в состав агента. Для этой цели используется оператор CREATE SLOT, имеющий следующий общий формат:

```
CREATE SLOT slot-name {{OBG}, {CON}, {ACT}, {STA}, {PROC}}
```

В результате выполнения этого оператора будет создан слот-отношение, имя которого определяется параметром slot-name. Набор типовых атрибутов [OBG], [CON], [ACT], [STA], [PROC] определяет функциональное назначение слота-отношения. Формат типовых атрибутов:

OBG [character], <определяет путь к нахождению объекта>;

CON [character], <содержит условие выполнения операции, указанной во фразе ACT, над объектом, путь к которому содержится в OBG>;

ACT [character], <определяет тип операции над объектом OBG>;

STA [character, 1|0|IS NULL], <содержит признак завершения операции над объектом OBG>;

1 — операция завершена успешно;

0 — операция завершена аварийно;

IS NULL — операция находится в стадии выполнения>;

PROC [character], <расширение опции CON, определения условий выполнения операций над объектами >; slot-name, <ключевой атрибут отношения AGENT [name]>.

С течением времени структура программного агента может изменяться, для удаления слота-отношения можно применять оператор, имеющий следующий формат:

```
DROP SLOT slot-name [RESTR|CASCAD]
```

Ключевое слово RESTR (принимается по умолчанию), при этом связи слотов-отношений со слотами-отношениями других программных агентов должны быть отключены, иначе выполнение операции будет отменено. Ключевое слово CASCAD определяет автоматическое удаление всех слотов-отношений других программных агентов, связанных с данным слотом-отношением.

5. Пример

Рассмотрим пример логической модели программного агента с именем CONTROL, который выполняет две задачи и, соответственно, состоит из двух слотов.

Слот A1 содержит четыре атрибута. Первый атрибут определяет объект действия — модуль провер-

ки вводимого логина и пароля password.exe, который находится в каталоге c:\MP. Второй атрибут определяет условие, при котором должно состояться выполнение операции, — если текущее время удовлетворяет ограничениям Time > 8-00 and Time < 18-00. Третий атрибут определяет вид операции или вид действия — запуск приложения password.exe. Четвертый атрибут определяет уровень приоритета — очень важно.

Слот A2 также содержит четыре атрибута и выполняет задачу активизации модуля model.exe, в котором программно реализованы все функции по построению математической модели почерка пользователя. Эта задача выполняется только после успешного завершения предыдущей задачи A1. Структура программного агента, выполняющего рассмотренные выше функции, приведена в табл. 2.

Таблица 2

Логическая структура программного агента CONTROL

Имя слота	Объект	Условие	Действие	Приоритет
A1	OBG: c:\MP\ password. exe	CON: IF 00<Time< 18-00	ACT: RUN c:\MP\ password. exe	STA: VERY IMP
A2	OBG: c:\MP\ model. exe	CON: AFTER A1	ACT: RUN c:\MP\ model.exe	STA: IMP

Реализация предлагаемого способа идентификации слушателей в системе дистанционного образования позволит повысить эффективность мер авторского контроля обучения.

Выводы

В статье рассмотрены вопросы построения модели пользователя в системе дистанционного образования. Предложен современный подход к идентификации пользователей на основе технологии программных агентов, комплекс языковых средств создания программного агента, разработана логическая схема интеллектуального программного агента, приведен пример логической схемы агента.

Список литературы: 1. Simon A. R. Strategic Database Technology: Management for the Year 2000. — Morgan Kaufmann Publishers, 1995 ISBN 1-55860-264-X, pp. 100–250. 2. Wooldridge M. and N. Jennings (1995) Intelligent agents: theory and practice. The Knowledge Engineering Review, 10(2), pp. 115–152. 3. Intelligent Agents II / Ed. M. Wooldridge, J. P. Mueller, and M. Tambe. — Springer-Verlag, 1996. — Lecture Notes in Artificial Intelligence, — V. 1037, p. 356. 4. Пономаренко Л. А., Філатов В. О. Програмні агенти технології в адмініструванні баз даних. // Вісник Київського торговельно-економічного університету. — К. — Вип. 3/2001. — С. 68–73.

Поступила в редколлегию 15.04.07