

ций должен дополняться иным подходом к отбору содержания лабораторных занятий. Здесь уместно оперировать теми парадигмами науки, которыми руководствуются исследователи в отборе и изучении проблем, а также в оценке и критическом анализе их решения.

Продуктивность этого подхода, как показала практика преподавания и проведенные авторами доклада педагогические исследования, обеспечивается особой педагогической технологией. При этом под педагогической технологией понимается комплексная интегративная система, включающая упорядоченное множество операций и действий, обеспечивающих педагогическое целеопределение, содержательные информационно-предметные и процессуальные аспекты, обеспечивающие направленное усвоение знаний, приобретение профессиональных умений и формирование личностных качеств студентов.

Отбирая парадигмы как материал для выработки конечных целей обучения, преподаватель стоит в начале пути, который должен привести студентов к сосредоточению их внимания на необходимости определения для решения конкретной проблемы своего рода общего знаменателя в значительном разнообразии знаний. Иными словами, студенты должны научиться находить решения проблемы, опираясь на структуру содержания учебной дисциплины, которая должна быть аналогична структуре современной научной информации и методов ее переработки.

УДК 681.3

ЛОГИЧЕСКИЙ ВЫВОД В ДИАЛОГОВОЙ ОБУЧАЮЩЕЙ СИСТЕМЕ

*Н. И. Кузьмицкий, В. И. Хведчук,
Брестский государственный технический
университет,
Брест, Беларусь*

Одной из задач при составлении обучающих и контролирующих курсов является генерация вариантов заданий. Обычно она решается автором таких курсов без привлечения дополнительных средств автоматизации. Вместе с тем данный процесс занимает достаточно много времени. При этом все более возрастают требования к сокращению времени на разработку курсов. Это вызвано

необходимостью освоения новых предметных областей, их быстротекущим изменением [1]. Вместе с тем данный процесс во многом является рутинным и мог бы быть реализован автоматизированным способом. Поэтому одним из требований к обучающей системе является наличие средств автоматизации генерации заданий.

1. ОБЗОР СУЩЕСТВУЮЩИХ ОБУЧАЮЩИХ СИСТЕМ

Известны следующие группы технологий разработки компьютерных курсов. К первой относятся технологии создания гипермедийных приложений. Они реализуются на базе систем Author Ware Prof, HyperCard, Course Builder, HM-Card. К второй — технологии с использованием инструментальных систем разработки компьютерных обучающих программ таких как РАКЕЛЬ, АДОНИС, УРОК, АОСМИКРО, СЦЕНАРИЙ. К наиболее перспективным относятся средства третьей группы, использующие сетевые технологии. Средствами разработки сетевых курсов являются WebCT, ToolBook II, ClassWare, Nicenet [2]. В качестве средств разработки систем дистанционного обучения рассматриваются Convenc, First Class Collaborative Classroom, Lotus Learning Space, Pla@d, MentorWare, WebMentorEnterprise [3]. Возможно использование для образовательных целей и прикладных научных пакетов MathCad, MatLab, Maple, Derive и др. Это объясняется прежде всего их мощными вычислительными и графическими возможностями. Но затраты по усвоению содержательной части изучаемой дисциплины сравнимы с затратами времени на освоение системы. Поэтому выделяется также использование предметно-ориентированных систем, таких как СПЕКТР, учебных пакетов ФОРМУЛА, МАТРИЦА, систем моделирования Model Vision Studium, Click'n, Analog Connection Workbench, Interactiv Physics. Данные системы отличаются максимальной адаптацией для использования в учебном процессе. Наиболее распространенной отмечается технология прямого программирования, использующая алгоритмы предметной области. В этой технологии используются такие средства как C++ Builder, Delphi, Visual C++, Visual Basic и др. В качестве наиболее перспективной рассматривается технология прямого кодирования на базе CASE-систем. Для получения наибольшего эффекта в них используется язык описания предметной области. Система СФИНКС позволяет создавать авторские алгоритмы на основании такого описания. В каче-

стве языков программирования такого рода систем используются языки визуального программирования Prograph, CODE 2. 0, VEX, Form/3 и др., используемые в системах Prograph, Create, Insecta Flow Coder и др [2].

Недостатком CASE-систем, типа СФИНКС, для разработки обучающих курсов является специализация в отдельной предметной области. Поэтому была поставлена задача разработки обучающей системы, позволяющей ее переориентацию на другие предметные области.

2. Представление обучающего и контролирующего курсов

Для представления обучающих курсов используются аппарат сетей Петри [4,5], графовые программные грамматики [6].

Обучающий курс в рассматриваемой системе представляется в виде множества

$$OK=U, M, Ru, Rm,$$

где Ru — множество отношений, заданное на множестве утверждений U , Rm — множество соответствий элементов множества мультимедийных подсказок M элементам множества U .

Контролирующий курс представляется в виде

$$KK=V, M, R, L, A, Rv, Rmv, Ra, Rl,$$

где Rv — множество отношений, заданное на элементах множества вопросов V , Rmv — множество отношений, задающих соответствие элементов множества M элементам множества V , Ra — множество отношений, задающих соответствие элементов множества ответов A элементам множества V , Rl — множество соответствий элементов множества оценивания L элементам множества V . Элементы множеств V , U , A , L представляют собой строки, в которых хранятся элементы обучающего и контролирующего курсов. Имеется возможность иерархического объединения элементов множеств U , L , A , V при помощи отношений Ru , Ra , Rl , Rv , Rmv . Каждому из элементов множеств Ru , Rl , Ra , Rv , Rmv может быть сопоставлен идентификатор, отражающий отношения иерархической группировки элементов V , U , L , A , M .

3. Средства логического вывода

Одним из способов сокращения затрат на разработку тестовых и обучающих курсов является генерация соответствующих последовательностей

в процессе диалога с обучаемым. Одним из средств генерации является логический вывод.

Для реализации логического вывода в диалоговой обучающей системе представим обучающий (контролирующий) курс в виде:

$$R1(a^1_1, a^1_2, \dots, a^1_{n1}), R2(a^2_1, a^2_2, \dots, a^2_{n2}) \dots Rm(a^m_1, a^m_2, \dots, a^m_{nm})$$

где a_{ij} — i -ый факт (вопрос) из j -го отношения R_j . Логическим выводом над данным отношением будем называть следующую операцию:

$$R_{k1}, R_{k2} \dots R_{kp} \rightarrow R_{kv}$$

В качестве отношения может быть группа фактов (вопросов). Аргументами факта могут быть переменные и константы, т. е. факт может быть представлен в виде:

$$a(l_1, l_2, \dots, l_s)$$

Например, факт в виде:

Человек(Сократ).

Если вводимая переменная имеет фиксированный набор значений по которым предусмотрен фиксированный набор действий, то возможно:

Если Сократ человек, то переход к вершине C ;

Если Конфуций человек, то переход к вершине C ;

Если переменная принимает иное значение, то переход к вершине D .

Вершина может быть C в виде факта:

Смертен(человек).

При этом возможен вывод:

Если Сократ человек, то человек смертен;

Если Конфуций человек, то человек смертен.

Возможны также ситуации S , зависящие от нескольких параметров:

$$S(p_1, p_2, \dots, p_l)$$

Они могут быть преобразованы к ситуации с одним параметром с помощью операции:

$$S(p_1, p_2, \dots, p_l) \rightarrow (S^{c1}(p^{c1}_1, p^{c1}_2, \dots, p^{c1}_l), S^{c2}(p^{c2}_1, p^{c2}_2, \dots, p^{c2}_l) \dots S^{cz}(p^{cz}_1, p^{cz}_2, \dots, p^{cz}_l))$$

Каждая из ситуаций S^{ct} имеет только один параметр, остальные выступают в виде констант.

Заключение

Возможно использование такого подхода при описании ситуаций в различных сферах деятель-

ности. Одним из применений может быть описание экономической ситуации.

В настоящее время используются в учебном процессе в Брестском государственном техническом университете на кафедре информатики и прикладной математики отдельные элементы диалоговой обучающей системы. Их использование показало возможность сокращения времени преподавателя на контрольный опрос, позволило сократить время на освоение новых тем.

Литература

1. Буза М. К., Дубков В. П., Зимянин Л. Ф. Концептуально-логическая схема совершенствования курсов по информатике // В сб. тр. Межд. конф. Сетевые компьютерные технологии. 25–29. 10. 2000. с. 142–153.

2. Степанов Д. Ю. Технология разработки компьютерных курсов по математическим дисциплинам в инструментальной CASE-системе СФИНКС // Информационные технологии. 2001. N 5. с. 42–51.

3. Змитрович А. И., Меуег А. О дистанционном обучении // В сборнике тр. Межд. конф. Сетевые компьютерные технологии. 25–29. 10. 2000. с. 161–165.

4. Пантелеев Е. Р., Ковшова И. А., Малков И. В., Пекунов В. В., Первовский М. А., Юдельсон М. В. Среда разработки программ дистанционного обучения ГИПЕРТЕСТ: инструментальные средства // Информационные технологии. 2001. N 8. с. 34–40.

5. Пантелеев Е. Р. Среда разработки программ дистанционного обучения ГИПЕРТЕСТ: логическая модель и архитектура // Информационные технологии. 2001. N 5. с. 30–36.

6. Певзнер Л. В. Гипертекстовая структура компьютерного учебника по информатике. // В сборнике тр. Межд. конф. Сетевые компьютерные технологии. 25–29. 10. 2000. с. 154–157.

УДК 681.3

СИСТЕМА ОБЕСПЕЧЕНИЯ КАЧЕСТВА

*В. В. Бахтизин, Д. Г. Круглов,
Белорусский государственный университет
информатики и радиоэлектроники,
Минск, Беларусь*

Lotus Notes/Domino — уникальная платформа, способная в минимальные сроки и с минимальными затратами автоматизировать большинство рабочих процессов современного предприятия. Однако, без системы обеспечения качества, по прошествии нескольких лет, Domino может превратиться в источник постоянных проблем. Главной причина такой трансформации является возможность разработчика быстро передать пользователю программное средство (ПС), не составляя техническое задание (ТЗ), пропустив этап проектирования, программируя задачу без учета каких либо стандартов и завершив разработку без тестирования. При этом за счет надежности самой платформы, большую часть времени программа будет работать без ошибок. Именно эта особенность Domino так притягательна для большинства пользователей. Однако, множество организаций, для которых эта особенность была главной причиной перехода на Domino через год или два сталкивались с тем, что они теряют контроль над информационной системой из сотен некачественных программ.

Потеря контроля над информационной системой доказывает, что проектирование, разработка, внедрение и сопровождение программ в среде Domino должны вестись с применением системы обеспечения качества ПС (SQA — software quality assurance). Необходимость в системе SQA в среде Domino также велика, как, например, при разработке баз данных на SQL или приложений на C/C++. Повышение скорости разработки, упрощение поддержки и обслуживания, продление срока службы приложения и облегчение последующей модернизации — основные задачи системы SQA актуальны и для Domino.

В последние годы интерес к подобным системам усиливается, особенно в связи с предстоящим крупным обновлением платформы Domino, и связанными с этим задачами разработки новых и миграции существующих приложений. Этим обусловлено появление инструментов, частично автоматизирующих отдельные этапы жизненного цикла