

6. Импорт/экспорт графики и текста

ИнГИС имеет встроенный модуль преобразования распространенных текстовых и графических форматов во внутренний, и наоборот. В качестве таковых для данной системы определены форматы DBF (для текста) и DXF (для графики).

Перечисленные функциональные возможности ИнГИС позволяют использовать ее в качестве удобного отечественного инструментального средства, обеспечивающего оперативную реализацию ГИС-проектов для различных приложений и с различными уровнями функциональных возможностей: от справочной ГИС до системы диспетчеризации и ситуационного управления технологическими процессами конкретного объекта.

Примерами таких ГИС-проектов могут быть: муниципальные ГИС для различных сегментов городского хозяйства (недвижимость, инженерные сети, земельные ресурсы и др.), электронные кадастровые системы (комплексные кадастры регионов), ГИС-предприятий с возможностями диспетчеризации и управления режимами работы вспомогательного энергетического хозяйства и основными технологиями.

Программная система разработана в среде Borland Delphi и предназначена для эксплуатации под управлением 32-разрядной ОС WINDOWS NT (WINDOWS-95) на ПЭВМ не ниже: 486 DX4, ОЗУ – 8 Мб, SVGA, с сетевым оборудованием.

Литература: 1. Гради Буч. Объектно-ориентированное проектирование с примерами применения/Пер. с англ. К.: Диалектика, 1992. 519 с. 2. Программно-аппаратное обеспечение, фонд цифрового материала, услуги и нормативно-правовая база геоинформатики. Ежегодный обзор. М: ГИС-ассоциация, 1996. Вып. 2 (1995). 325 с.

Поступила в редколлегию 22.03.98

Ткаченко Владимир Филиппович, канд. техн. наук, профессор, заведующий кафедрой ИКГ ХТУРЭ. Адрес: 310166, Украина, Харьков, ул. Акад. Ляпунова, 11, кв. 25, тел. 40-93-78, 45-27-72.

Бабийчук Станислав Викторович, старший научный сотрудник кафедры ИКГ ХТУРЭ. Адрес: 310180, Украина, Харьков, ул. Отакара Яроша, 17б, кв. 45, тел. 40-93-78, 32-84-50.

Губа Николай Иванович, доцент кафедры ИКГ ХТУРЭ. Адрес: 310093, Украина, Харьков, ул. Скорохода, 24, кв. 56, тел. 40-93-78, 72-02-12.

Тришин Владимир Вячеславович, научный сотрудник кафедры ИКГ ХТУРЭ. Адрес: 312154, Украина, Харьковская область, Харьковский район, пос. Буды, ул. Октябрьская, 86, тел. 40-93-78.

Челомбителько Виктор Федорович, старший преподаватель кафедры ИКГ ХТУРЭ. Адрес: 310075, Украина, Харьков, ул. 3-го Интернационала, 11, кв. 67, тел. 40-93-78, 94-26-85.

УДК 519.687

НЕКОТОРЫЕ АСПЕКТЫ ПРИМЕНЕНИЯ И КОНСТРУИРОВАНИЯ КОМПЬЮТЕРНЫХ ОБУЧАЮЩИХ СИСТЕМ

ДЮБКО Г.Ф., СЕРДЮЧЕНКО В.Я., СОЛОВОВА С.А.

Рассмотрены некоторые аспекты применения компьютерных обучающих систем (КОС), касающиеся повышения эффективности учебного процесса, проводимого с применением КОС, а также снижения трудоемкости их разработки. Приведен пример использования интеллектуальных возможностей КОС при контроле знаний.

Применение компьютера в обучении – проблема многогранная и комплексная. Компьютер предоставляет как обучаемому, так и обучающему новые возможности, которые реализуются путем создания программных продуктов, имитирующих изучаемые процессы и сам процесс обучения. Поэтому компьютерное обучение необходимо рассматривать с различных точек зрения: методической, программной, прикладной.

Двумя основными и взаимосвязанными аспектами компьютерного обучения являются методика использования КОС, которая предъясвляет те или иные требования к программному продукту, и технология создания КОС, результатом которой явля-

ется программный продукт. Рассмотрим эти два аспекта подробнее.

Одной из зарекомендовавшихся форм КОС являются электронные учебники. В настоящее время выработалась структура такого учебника: оглавление с возможностью выбора нужного материала; раздел обучения, где обучающийся может пассивно (путем чтения) или активно (путем некоторых манипуляций) изучать учебный материал; самоконтроль, позволяющий обучаемому определить для себя степень усвоения материала; контроль – выполнение контрольной работы с выставлением оценки. Существенным является то, что в разделе обучения учебный материал подается небольшими порциями. Каждая порция должна представлять определенную сущность изучаемого предмета. После изучения такой сущности обучаемый отсылается в режим самоконтроля, где ему предлагаются упражнения по использованию данной сущности в различных употреблении. При ошибочном ответе самоконтроль, в зависимости от типа ошибки, отсылает обучающегося к повторению соответствующей порции учебного материала с определенным комментарием и рекомендациями.

Создание электронного учебника как в методическом плане, так и в плане программной реализации – работа весьма трудоемкая. В существующих системах создание электронного учебника требует тесного сотрудничества программиста и специалиста в предметной области. Наиболее трудоемкой операцией является связывание текста изучаемого материала с самоконтролем и возможные переходы между различными режимами.

При изучении любой дисциплины с компьютерным учебником или без него главной фигурой в обучении является преподаватель, а основным отношением в преподавании есть обратная связь от обучаемого к преподавателю. Такая связь осуществляется путем фиксации ошибочных действий обучаемого, которые характеризуют его степень непонимания. Чтобы осуществить такую обратную связь, необходимо иметь базу данных результатов контроля (БДРК) – некоторый аналог журнала успеваемости с расширенными возможностями. Пользуясь БДРК, преподаватель может оперативно вмешиваться в процесс преподавания для каждого учащегося индивидуально. В этом случае режимы контроля и самоконтроля должны иметь интерфейсы с БДРК и фиксировать там действия учащегося.

Как показывает практика, преподаватели неохотно используют электронные учебники в учебном процессе, чаще всего рекомендуя их для самостоятельной работы. Зачастую, на наш взгляд, это происходит от того, что учебник составлен не по методике, которой руководствуется преподаватель, или преподаватель хотел бы изменить последовательность подачи материала, порции материала, но не может этого сделать без участия программиста. Поэтому электронный учебник рационально дополнить средствами, позволяющими преподавателю свободно манипулировать учебным материалом во всех разделах обучения – средой подготовки занятий (СПЗ). Пользуясь СПЗ, преподаватель вообще может сам, без программиста, составлять свои компьютерные уроки. Очевидно, что СПЗ также является средством уменьшения трудозатрат при составлении самого электронного учебника. Электронный учебник, дополненный СПЗ, можно использовать для подачи нового материала, для комбинированного использования компьютерных средств мультимедиа с учебной доской. Такая система больше соответствует названию КОС. Если КОС для какой-то конкретной дисциплины снабдить базой данных предмета, включив в нее ключевые сущности, рисунки, анимации и подключить эту базу к СПЗ, трудозатраты на подготовку занятия еще более уменьшатся.

При подаче нового материала, самостоятельном изучении, информационном поиске наиболее продуктивно использование КОС в сетевом варианте. Представляется разумным строить сетевые варианты КОС на основе технологии широко используемых сетей, таких как Internet и ее локального варианта – Intranet.

Одной из ключевых проблем применения КОС является проблема контроля, поскольку именно в этом разделе происходит обратная связь от ученика к учителю. Сейчас доминирует тестовая система контроля, которая весьма эффективна в плане временных затрат ученика для задач, решаемых в один-два шага. Обычно применяется одноуровневая система тестового контроля. Здесь не нужно что-то писать, излагать, выводить. Достаточно лишь нажать клавишу, выбрав правильный ответ.

Однако там, где для решения задачи требуется большое количество шагов, и в особенности там, где нужно контролировать логику, сам процесс рассуждения, тестовый контроль становится неэффективным. Можно, конечно, предложить систему многоуровневого тестового контроля, где каждый уровень

соответствует шагу решения задачи. Однако такой подход требует от преподавателя больших трудозатрат по составлению тестов и часто также неэффективен.

Например, при изучении программирования одним из наиболее трудно усваиваемых разделов является алгоритмизация, где по словесному описанию проблемы необходимо составить схему алгоритма, пользуясь каким-либо графическим ее представлением. Пусть из двух подходов, структурного и объектно-ориентированного, изучается структурный. В качестве схематического представления изучаются программные деревья, что требует разбиения исходной задачи на подзадачи и декомпозицию подзадач до уровня трех операторов: следования, условного, повторения. Как известно из теоремы Бозма-Джакопини, любая программа может быть представлена комбинацией этих трех операторов.

Положим, что после изучения алгоритмизации с помощью программных деревьев необходимо контролировать степень усвоения материала не только с точки зрения правильности конечного результата, но и с точки зрения подхода к построению дерева, корректности декомпозиции исходной задачи на подзадачи, к стилю создания дерева. Рассмотрим эти вопросы на конкретном примере.

Упражнение 1. Составить алгоритм в виде программного дерева для нахождения наибольшего общего делителя (НОД) двух натуральных чисел, одно из которых есть максимальный (MAX) элемент последовательности натуральных чисел, заканчивающейся символом "-1", а другое – минимальный (MIN) элемент этой последовательности. Найти НОД, используя алгоритм Евклида.

В качестве тестов для ответа на это упражнение можно предложить готовое программное дерево и его варианты с различными "осмысленными" ошибками. Однако такой подход малоэффективен, так как обучающийся, не построив сам требуемое дерево, мало получит для себя из этого упражнения и даст мало информации о себе преподавателю. В данном случае при тестовом подходе лучше устроить многоуровневый контроль. Здесь можно предложить такие уровни:

1. Разбивка на подзадачи (MIN, MAX, НОД).
2. Определение инвариантов (переменных) подзадач и задачи (min, max, ТЕК_Е – текущий элемент).
3. Определение последовательности выполнения подзадач (MIN, MAX, НОД или MAX, MIN, НОД).
4. Идентификация самого дерева.

При многоуровневом контроле обучающийся пропускается на следующий уровень только после выполнения заданий предыдущего уровня. Здесь уже можно в какой-то мере контролировать подходы к решению задачи, выбор стратегии и так далее. Однако трудоемкость подготовки многоуровневых тестов преподавателем очень велика.

Значительный объем информации об обучаемом можно получить, предоставив ему возможность излагать решение задачи в привычной форме – на языке предметной области. Но в этом случае КОС должна сама обладать интеллектуальными способностями, что приводит к использованию экспертной системы (ЭС).

При конструировании ЭС мы исходим из концепции, что ЭС должна обладать не только знаниями о решении задачи, но и знаниями преподавателя, проверяющего решение. Поскольку решение задачи обучающимся постулируется в виде последовательности языковых конструкций (ЯК) некоторого языка, ЭС должна понимать этот язык, чтобы формулировать имеющиеся факты, и на их основе проверять правильность решения или давать рекомендации в ходе самого решения. При таком подходе к конструированию ЭС основными ее компонентами становятся языковой процессор (ЯП) и базы знаний (БЗ).

ЯП расшифровывает смысл, зашифрованный в ЯК, представляющей шаг решения задачи. Смысл некоторой ЯК – это сущности, их свойства, отношения между сущностями, которые присущи рассматриваемой предметной области. Исходя из смысла ЯК и соответствующих БЗ, выбирается стратегия решения задачи и конкретные знания, необходимые для выполнения шага решения. Использование языка предметной области в качестве входного ЯП требует анализа языка, являющегося некоторым формализованным подмножеством естественного языка. Поэтому ЯП должен анализировать естественно-языковые конструкции (ЕЯК), превращая их в смысл. Для пояснения сказанного выше обратимся к упражнению 1.

Чтобы сократить объем рассматриваемой информации, сократим формулировку упражнения 1 до фразы "Составить алгоритм в виде программного дерева для нахождения наибольшего общего делителя двух натуральных чисел" и будем считать эту фразу входной ЕЯК языкового процессора. Выполнив анализ языковой конструкции, ЯП должен выдать результат в виде семантической функции

$$g_1(x_0, x_1, x_2, x_3), \quad (1)$$

где g_1 – глагольная семантическая функция, задающая действие композиции некоторой сущности; x_0 – глагол со всеми значениями его атрибутов; x_1 – композируемая сущность; x_2 – форма (вид), в которой должна быть представлена сущность x_1 ; x_3 – назначение (использование) сущности x_1 .

В свою очередь, x_1, x_2, x_3 сами являются семантическими функциями. Так, x_1 – это семантическая функция

$$V_j \text{ (алгоритм)}.$$

Здесь V_j указывает значение (V – значение, j – номер этого значения) слова "алгоритм" в "толковом словаре", используемое именно в анализируемой фразе, x_2 выражается семантической функцией

$$P_k(V_{j1}(в), S_{k1}(V_{j2}(виде), A_{k2}(V_{j3}(дерева), V_{j4}(программного))))),$$

где P_k – предложная функция; S_{k1} – функция родительного падежа существительного; A_{k2} – функция прилагательного.

Каждая из семантических функций индексирована. Например, в рассматриваемой фразе предлог "в" употребляется в значении p_4 – четвертая позиция предлога в словаре русского языка С.И. Ожегова, которая гласит: "Употребляется при указании на внешний вид кого-, чего-нибудь, на оболочку, на одежду". Таким образом, формула (1) разворачивается в суперпозицию семантических функций.

Кроме того, мы должны дополнить входной язык средствами представления схем программ. Каждая входная ЕЯК должна отображаться в "толковом словаре". Этот словарь можно трактовать как лингвистико-смысловую базу знаний (ЛСБЗ). ЛСБЗ играет значительную роль при анализе исходных ЕЯК. Организация ЛСБЗ аналогична организации толкового словаря естественного языка с добавлением трактовки смысловых функций для словосочетаний. В ЛСБЗ каждое слово представлено значениями, в которых расшифровывается смысловое употребление этого слова. Значение сформировано в виде суперпозиции семантических функций, что позволяет формальную манипуляцию на ЛСБЗ.

Для выполнения автоматического анализа исходной ЕЯК предлагается использовать синтаксический анализатор в атрибутивных транслирующих грамматиках специального типа.

Таким образом, исходный язык предметной области будет описываться транслирующей атрибутивной грамматикой специального вида, представляющей собой синтаксическую базу знаний, и ЛСБЗ. Предложены алгоритмы синтаксического анализа, преобразовывающие исходную языковую конструкцию в суперпозицию семантических функций. Использование в алгоритмах ЛСБЗ позволяет делать детерминированный семантический выбор в большинстве случаев. Однако говорить, что вопрос анализа ЕЯК решен, нельзя. Детерминировать семантику более точно при достаточном объеме входной информации и информации в БЗ можно только при использовании знаний о предметной области в полном объеме. Рассмотрим теперь более подробно вопрос о представлении БЗ предметной области. Представление знаний может быть дано в различной форме. Основными формами являются логическая, сетевая и объектная. В целях эффективности эти формы объединяются в смешанное представление. В ЭС для КОС используется представление знаний в виде связанной системы таблиц решений. Однако ниже мы рассматриваем логическую форму представления знаний, на основе которой может быть сформулирована любая другая.

Проиллюстрируем принципы, заложенные в методику составления и использования предметной БЗ, на примере. Для этого вернемся к упражнению 1 и выделим в нем фрагменты нахождения максимума указанного списка элементов. Алгоритм в форме программного дерева может иметь вид схемы (рисунок).

Положим, что ответ учащегося на вопрос построения соответствующего алгоритма должен состоять из списка задач, подзадач, инвариантов задачи и программного дерева типа рисунка. (Можно ограничиться только этим деревом.) Чтобы проконтролировать правильность ответа, учитывая рассуждения и корректность построения, нужна предметная база знаний. Такая база знаний может быть представлена в логической форме следующими правилами (предикатами):

$$\text{МЕНЬШЕ}(\beta_1, \beta_2) \wedge \text{ПРИСВ}(\beta_1, \beta_2) \wedge \text{ПОСЛЕД2}(\text{'меньше', 'присвоить'}) \supset \text{IF2}(\beta_1, \beta_2) \quad (2)$$

$$\text{НЕРАВНО}(\beta_2, \gamma) \wedge \text{IF2}(\beta_1, \beta_2) \wedge \text{INIT}(\beta_2) \wedge \text{ПОСЛЕД3}(\text{'неравно', 'if', 'init'}) \supset \text{ПОКА2}(\beta_2, \gamma) \quad (3)$$

$$\text{ПОКА}2(\beta_2, \gamma) \wedge \text{ПОКА}(\beta_2, \gamma) \supset \text{ПОДЗМАХ}2(\alpha_2, \beta_2, \gamma) \quad (4)$$

$$\text{INIT}(\beta_1) \wedge \text{INIT}(\beta_2) \supset \text{ПОДЗМАХ}1(\alpha_1, \beta_1, \beta_2) \quad (5)$$

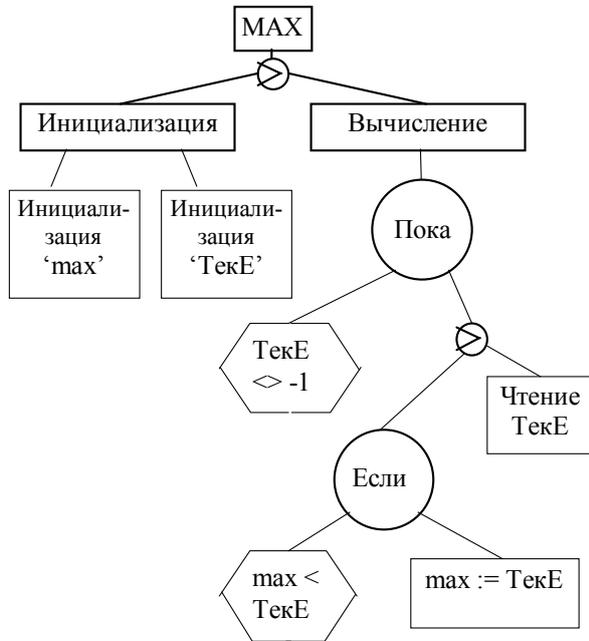
$$\text{ПОДЗМАХ}1(\alpha_1, \beta_1, \beta_2) \wedge \text{ПОДЗМАХ}2(\alpha_2, \beta_2, \gamma) \wedge$$

$$\text{ИМЯПОДЗ}(\alpha_1, \text{'первая'}) \wedge \quad (6)$$

$$\text{ИМЯПОДЗ}(\alpha_1, \text{'вторая'}) \supset \text{ВЕРНО}(\alpha_1, \alpha_2, \beta_1, \beta_2, \gamma)$$

$$\text{ВЕРНО}(\alpha_1, \alpha_2, \beta_1, \beta_2, \gamma) \supset \quad (7)$$

$$\text{ЗАДАЧАМАХ}(\alpha_1, \alpha_2, \beta_1, \beta_2, \gamma) .$$



Программное дерево для нахождения максимума среди элементов списка

Формулы (2) – (7) можно трактовать как предметную базу знаний для построения алгоритмов вычисления максимума среди элементов списка, заканчивающегося маркером. Элементарные предикаты, входящие в состав этих формул, отражают наши знания о программных деревьях:

МЕНЬШЕ (β_1, β_2) — $\beta_1 < \beta_2$ (не в смысле вычисления, а в смысле существования такой языковой конструкции);

ПРИСВ(β_1, β_2) — $\beta_1 := \beta_2$;

ПОСЛЕД2(x_1, x_2, \dots, x_n) — x_1 предшествует x_{i+1} ;

НЕРАВНО(β_2, γ) — $\beta_2 < \gamma$;

ИМЯПОДЗ($\alpha_1, \text{'первая'}$) — указывает, что строка α_1 , показывающая имя подзадачи, является первой подзадачей.

Подобные БЗ создаются для нахождения MIN и НОД, что даст полную БЗ для решения вопросов контроля решения задач типа упражнения 1. Эти базы знаний можно использовать в качестве подбазы в других сочетаниях знаний.

Заметим, что соответствующие элементарные предикаты с конкретными значениями своих термов создаются в качестве предикатов-фактов после обработки ответа учащегося языковым процессором. Также будут созданы предикаты-факты **МЕНЬШЕ**, **ПРИСВ**, **ПОСЛЕД2**, **ПОСЛЕД3**, **INIT**, **IF**, **ПОКА**, **ИМЯПОДЗ**, **ЗАДАЧАМАХ**, в которых будут указаны конкретные значения термов.

После обработки языковым процессором условия решаемой задачи и использования базы знаний, задающей стратегию решения задачи, определяется круг знаний, необходимый для решения задачи (по частям или в целом). По правилам из базы знаний и предикатам-фактам делается логический вывод, который и определяет правильность ответа. В зависимости от оставшихся при выводе (методом резолюций) предикатов выдается сообщение о конкретной ошибке учащегося. Таким образом, исходными данными системы контроля являются формулировка задачи и ответ учащегося; выходом — определение того, является ли ответ логическим следствием формулировки задачи. В настоящее время реализована программная оболочка, позволяющая осуществить основные режимы КОС с возможностями тестового контроля. Ведется работа по созданию программной оболочки, реализующей ЭС.

Поступила в редколлегию 28.01.98

Дюбо Генадий Федорович, канд. техн. наук, профессор кафедры ПО ЭВМ ХТУРЭ. Адрес: 310000, Украина, Харьков, пр. Гагарина, 38, кв. 70, тел. 27-13-26.

Сердюченко Виктор Яковлевич, канд. техн. наук, профессор кафедры ПО ЭВМ ХТУРЭ. Адрес: 310000, Украина, Харьков, пр. Перемоги, 57^Г, кв. 34, тел. 40-94-19.

Соловова Светлана Александровна, канд. техн. наук, старший научный сотрудник кафедры ПО ЭВМ ХТУРЭ. Адрес: 310000, Украина, Харьков, ул. Героев Сталинграда, 162, кв. 20, тел. 40-94-46.