

УДК 378.147; 004.94:62

А.Г. ЧУХРАЙ, Е.С. ВАГИН, Т.И. ШЕВЧУК

Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», Украина

МОДЕЛИ ПРЕДСТАВЛЕНИЯ ЗНАНИЙ В ИНТЕЛЛЕКТУАЛЬНЫХ КОМПЬЮТЕРНЫХ ПРОГРАММАХ, ОБУЧАЮЩИХ СХЕМОТЕХНИЧЕСКИМ РЕШЕНИЯМ

Выполнен обзор различных подходов к представлению знаний в существующих компьютерных системах обучения. Предложен авторский подход к представлению знаний предметной области – заданий по схемотехнике, основанный на разделении знаний на декларативные и процедуральные, а также позволяющий группировать их в иерархические деревья компонентов компетенции. На примере рассмотрен процесс генерации набора вариантов задания на основании предложенной модели представления знаний в области схемотехники. Описано авторское программное средство, облегчающее процесс создания подобных моделей.

Ключевые слова: компьютерные обучающие программы, алгоритмизация обучения, компонент компетенции.

Введение

В компьютерных системах обучения (КСО) обычно выделяются такие основные структурные компоненты: модель предметной области, модель студента, педагогическая модель и модель взаимодействия с пользователем. Модель взаимодействия с пользователем предоставляет средства коммуникации системы с пользователем. Педагогическая модель содержит набор стратегий обучения для внешнего обучающего цикла (выбор следующего задания в рамках обучающего курса) и внутреннего обучающего цикла (выбор следующего шага в рамках задания).

Принято считать КСО интеллектуальной (ИКСО), если педагогическая модель способна обеспечить студента релевантными его действиям и заданию подсказками [1]. Педагогическая модель использует знания и данные, представленные моделью предметной области и студенческой моделью. Представление знаний в КСО играет определяющую роль в формировании эффективных стратегий педагогической модели. В связи с этим разработка методов и средств моделирования знаний являются актуальными направлениями деятельности проектировщиков КСО. Поскольку такие средства направлены на описание знаний человека, желательно, чтобы их возможности были как можно выше. С другой стороны, если форма представления становится излишне сложной, то усложняется и механизм выводов, при этом не только затрудняется проектирование педагогической модели, но и возникает опасность потери достоверности выполняемых ею действий. В связи с этим исследователи в области ИКСО предлагают раз-

личные средства представления знаний. Перед формулированием задач исследования, сделаем краткий обзор наиболее известных средств.

1. Обзор подходов к представлению знаний в существующих обучающих системах и определение задачи исследования

Авторский инструментарий Cognitive Tutor Authoring Tools (СТАТ), разработанный в Университете Карнеги Меллон, Питтсбург, США, предоставляет средства для создания КСО двух типов: «Псевдо-наставник» и «Когнитивный наставник» [2]. Первый тип позволяет автору быстро создать пользовательский интерфейс и несколько сценариев решения задач. Второй тип позволяет добавить базу продукционных правил в визуальном редакторе (рис. 1а). Продукционная модель характеризуется наглядностью, высокой модульностью, легкостью внесения дополнений и изменений, простотой механизма логического вывода. Известным недостатком продукционной модели является неясность взаимных отношений между составляющими конкретную продукционную модель знаний правилами, а также правилами логического выбора.

Другая КСО, разработанная в Университете Ноттинггема, Великобритания, называется REDEEM [3]. Она позволяет структурировать знания, представляемые в виде учебных материалов, в соответствии с психофизиологическими особенностями различных групп студентов. Преподавателю предоставляются средства, при помощи которых он может описать учебный материал и стратегию обучения (рис. 1, б).

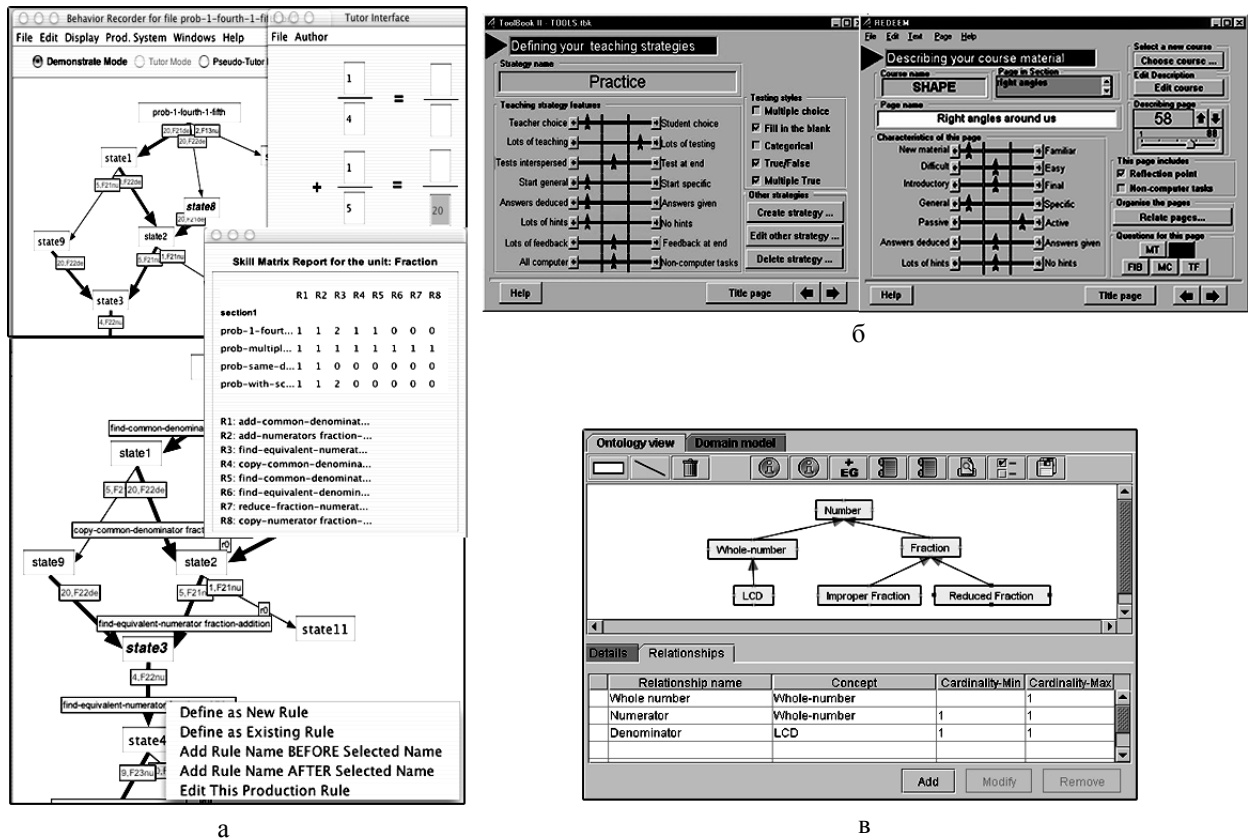


Рис. 1. Компьютерные средства моделирования знаний для ИККО: а – STAT; б – REDEEM; в – ASPIRE

Несмотря на наличие базы знаний (БЗ), поддержка диагностических моделей не предусмотрена. Адаптивная подсказка отсутствует, вместо этого формируется адаптивная последовательность изложения учебного материала.

ASPIRE – авторский инструментальный, созданный в Intelligent Computer Tutoring Group (ICTG) в Кентерберийском университете, Новая Зеландия (ранее разработка называлась WETAS) [4]. ASPIRE включает в себя инструменты для создания онтологии. Редактор онтологий (рис. 1в) позволяет представить знания в виде фреймовой модели предметной области, в которой каждый фрейм – важная для учебного задания сущность. Затем предлагается создать графический интерфейс для учебного задания и решить его несколькими способами. В результате предоставления нескольких решений ASPIRE генерирует семантические ограничения, используя которые, педагогическая модель дает подсказки обучаемому. Фреймовая модель наиболее эффективна, если родовидовые связи изменяются редко и предметная область содержит мало исключений. Другое достоинство фреймовой модели в том, что значения слотов (атрибутов сущностей) можно генерировать автоматически. Фреймовая модель имеет и недостатки. Основной из них – ее сложность, что сказывается на трудоемкости внесения измене-

ний и дополнений в иерархию сущностей, а также на скорости работы механизма вывода (для КСО – работе педагогической модели).

В трудах по ИККО часто упоминаются системы SOPHIE I, II, III, работающие в предметной области схемотехники. Это различные версии КСО, разрабатываемой изначально в Университете Калифорнии, Ирвин, США, известными учеными Brown и Burton на протяжении 1973-1977 годов [5]. Предметная область была сужена до единственного устройства – стабилизированного источника питания IP-28. Так как подсказки в SOPHIE генерировались при помощи деревьев принятия решений, было использовано множество дедуктивных правил, объединенных в иерархию. Часть дерева формировали правила, относящиеся к общим знаниям электротехники, а часть – правила, специфичные для IP-28. Возможность изменения и дополнения дерева в SOPHIE отсутствует. Перечисленные особенности SOPHIE позволили реализовать достаточно эффективную педагогическую модель, обладающую высокой точностью предоставляемых подсказок, возможностью автоматически идентифицировать неисправности, определять степень диагностируемости неисправностей по их признакам, однако со статической БЗ.

Наиболее преуспевшей в представлении знаний компьютерной обучающей системой, пожалуй, мож-

но считать Andes, в настоящий момент разрабатываемый в Государственном Университете Аризоны при поддержке Питтсбургского Центра Науки Обучения, (США) коллективом, возглавляемым Куртом ВанЛеном. Для представления знаний предметной области Andes использует так называемый «Граф решения» (англ. Solution Graph), который состоит из таких узлов: контекстно зависимые правила, факты, цели, узлы применения правил, узлы стратегий [6]. Педагогическая модель Andes использует граф решения для того, чтобы построить байесовскую сеть, используемую для того, чтобы формировать соответствующую поведению студента стратегию обучения. Вероятностный вывод позволяет Andes успешно генерировать подсказки в условиях частичной неопределенности знаний студента. Andes ориентирован на физические задачи и включает в себя несколько заданий по схемотехнике, что подтверждает эффективность выбранного подхода представления знаний для задач в этой области. Пожалуй, единственными существенными недостатками являются сложность составления графа решения и его ориентированность на конкретную задачу, а не на класс задач.

Перечисленные недостатки существующих систем и недостаточность средств представления знаний в области схемотехники говорит о необходимости разработки новых средств, способных уменьшить влияние существующих недостатков для КСО по схемотехнике.

2. Представление знаний в компьютерных программах, обучающих схемотехнике

Для решения задач по схемотехнике студент должен обладать определенной компетенцией. Будем называть компонентом компетенции (КК) некоторую структурную составляющую компетенции, применяемую в ходе решения задачи.

Например, в системе СТАТ, аналогом КК можно считать умения (Skills), являющиеся узлами определенного класса в дереве продукционных правил, а в Andes – части графа решения, завершающиеся узлами применения правил [6]. Воспользуемся декомпозицией знаний на декларативные и процедуральные [6,7]. Декларативные знания (ДЗ) являются понятийными представлениями, например, в области схемотехники – о характеристиках электрического сигнала, элементах электрической цепи и т.д. Процедуральные знания (ПЗ) – представления о том, каким методом (правилом) необходимо воспользоваться, чтобы получить из уже существующих ДЗ новые. В области схемотехники это может быть знание того, какое электронное устройство надо включить в цепь для получения на его выходе

сигналов с требуемыми характеристиками. Совокупность правильного применения ПЗ к верно выбранным ДЗ формирует КК. Так как применение ПЗ дает на выходе новые ДЗ, и они могут быть использованы для применения других ПЗ, то КК организовываются в иерархическую структуру. Это позволяет диагностировать ошибочные знания студентов, спускаясь вниз по иерархическому дереву. Например, если студент при поставленной задаче получить на выходе сигнал напряжения 3В при входном сигнале 5В, соединяет со входом делитель напряжения с необходимыми характеристиками, тогда с высокой вероятностью можно считать, что он владеет КК, связанным с делением напряжения. Если номиналы резисторов были выбраны неверно, тогда можно заключить, что владение КК частичное. Если было использовано другое устройство, но сигнал на выходе соответствует требуемым характеристикам, то владение КК подтверждается, но необходимо дополнительное исследование о его оптимальности. В двух последних ситуациях нужен переход на более глубокие уровни иерархии КК, например, путем определения значения силы тока, протекающего в цепи с двумя параллельно соединенными резисторами. Если же сложности возникают и здесь, то нужно спуститься еще на один уровень ниже. Стратегия обучения также может быть направлена с нижних уровней иерархии к верхним. Рассмотренный пример проиллюстрирован на рис. 2. Последовательность прохождения КК может быть определена исходя из структуры алгоритма решения задачи, схематично представленного в виде ДЗ (круглые блоки) и ПЗ (квадратные блоки). Такое представление будем называть моделью предметной области задания.

ДЗ могут быть представлены во фреймовом виде, так как имеют несколько характеристик (слов), таких как номинал, единицы измерения, величина погрешности, графическое обозначение и т.д. Это позволяет генерировать характеристики автоматически, увеличивая тем самым число вариантов задачи с одной и той же постановкой. ПЗ должны быть представлены в виде алгоритмов получения выходных ДЗ с использованием характеристик входных ДЗ, т.е. функционировать аналогично функциям в программировании.

В такой модели ДЗ могут быть предоставлены автором без связей между ними, как потребовалось бы при применении онтологий, что упрощает разработку БЗ. Отсутствует разделение ДЗ на подклассы (например такие, как контекстно зависимые знания, факты и цели в Andes), что упрощает создание модели предметной области задания и позволяет изменять направленность связей в модели, то есть использовать различные искомые и известные величины, рас-

ширя тем самым количество вариантов заданий, которые могут быть получены по созданной модели.

Работа программистов сосредоточена на написании алгоритмов для ПЗ. Таким образом, работа программиста и специалиста в предметной области разделяются. Автор обучающего задания выбирает отдельные ДЗ и ПЗ, определяет связи между ними

(рис. 2, 3). Это может быть реализовано в процессе решения задания в общем виде. КК ставятся строго в соответствие ПЗ, поэтому могут быть определены автоматически. Иерархическая организация КК при таком подходе к представлению знаний позволяет выбирать следующую задачу исходя из стратегии постепенного усложнения или облегчения заданий.

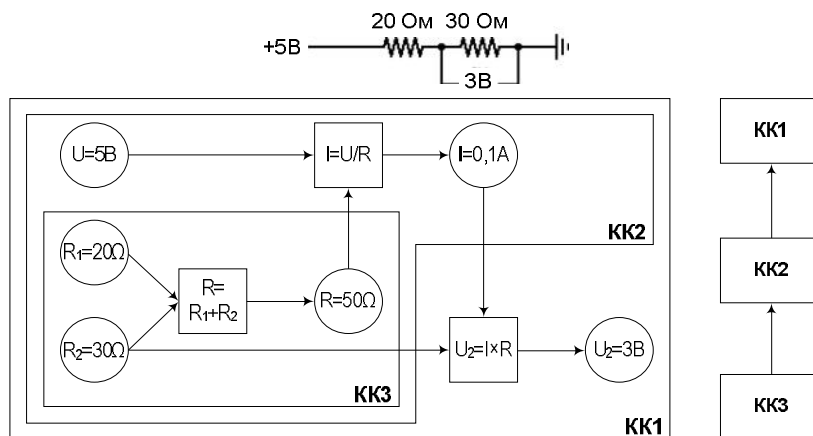


Рис. 2. Пример структуры КК для задачи с делителем напряжения

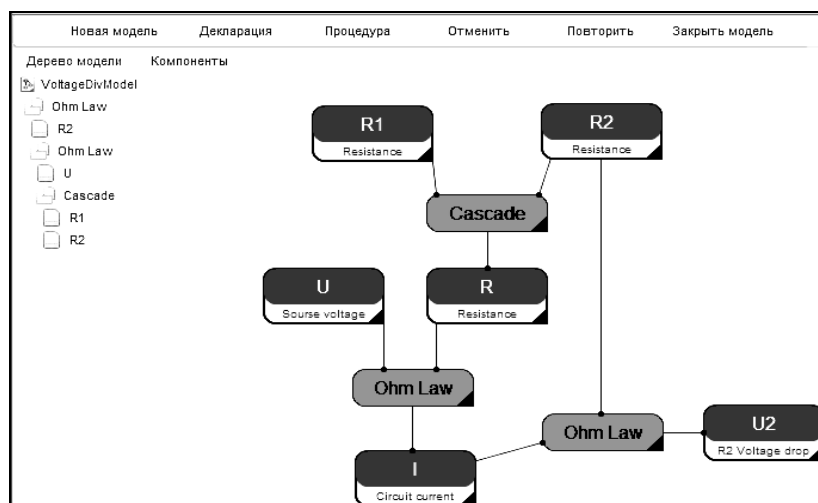


Рис. 3. Интрефейс компьютерного инструмента моделирования знаний предметной области задачи

Для одного типа КК могут быть предусмотрены различные алгоритмы получения выходных ДЗ при различных наборах входных ДЗ (КК1 и КК2 на рис. 2 имеют один тип). Перечисленные преимущества были выявлены в области схемотехники, однако могут быть обнаружены и в других предметных областях.

Представим модель, изображенную на рис. 2 в формализованном виде. Пусть D – множество всех ДЗ в модели задания ($D = \{R_1, R_2, R, U, U_2, I\}$). Обозначим ПЗ на применение закона Ома и вычисление сопротивления последовательно соединенных резисторов как P_{Ohm} и P_{Cas} . Эти ПЗ можно представить в виде функций:

$$P_{Ohm}(A) = \begin{cases} U/R & , \text{если } A = \{U, R\}, \\ U/I & , \text{если } A = \{U, I\}, \\ I \cdot R & , \text{если } A = \{I, R\}; \end{cases} \quad (1)$$

$$P_{Cas}(A) = \begin{cases} R_1 + R_2 & , \text{если } A = \{R_1, R_2\}, \\ R - R_1 & , \text{если } A = \{R, R_1\}, \\ R - R_2 & , \text{если } A = \{R, R_2\}. \end{cases} \quad (2)$$

Для равенств (1) и (2) справедливо $A \in D$.

Тогда решение задачи нахождения величины падения напряжения на резисторе R_2 , U_2 можно представить в виде выражения:

$$U_2 = P_{Ohm} \left(\left\{ P_{Ohm} \left(\left\{ P_{Cas} (R_1, R_2), U \right\} \right), R_2 \right\} \right). \quad (3)$$

Величины всех ДЗ, находящихся в правой части равенства (3) должны быть известны для выполнения условий существования и единственности решения. Обозначим множество этих ДЗ как V . Предположим, возникла ситуация, когда искомую величину необходимо изменить на номинал резистора R_1 . Изначально мы располагаем моделью $M = \{D, P, E\}$, где $P = \{P_{Ohm}, P_{Cas}\}$ – множество ПЗ, включенных в модель, а E – множество связей между ДЗ и ПЗ, определяемых выражениями (1) и (2). На первом шаге анализа модели необходимо определить ПЗ, при помощи которого можно вычислить значение номинала для R_1 . Исходя из (2) подходящее ПЗ – $P_{Cas}(A)$ при $A = \{R, R_2\}$. Номинал R может быть вычислен как при помощи $P_{Ohm}(A)$, так и при помощи $P_{Cas}(A)$, но во втором случае должно выполняться $A = \{R_1, R_2\}$, а R_1 является искомым ДЗ. Тогда $R = P_{Ohm}(A)$ при $A = \{U, I\}$. R_2 вычислить при помощи доступных в модели ПЗ нельзя, следовательно $R_2 \in V$. Аналогично исследуется возможность нахождения величин для ДЗ U и I . И так до тех пор, пока не будет получено окончательное выражение для нахождения номинала R_1 :

$$R_1 = P_{Cas} \left(\left\{ P_{Ohm} \left(\left\{ P_{Ohm} (U_2, R_2), U \right\}, R_2 \right) \right\} \right). \quad (4)$$

При получении выражений для некоторых ДЗ могут быть получены несколько путей нахождения искомой величины при различных значениях V . Выражения для других ДЗ из D :

$$\begin{aligned} R_2 &= P_{Cas} \left(\left\{ P_{Ohm} \left(\left\{ P_{Ohm} (U_2, R_2), U \right\}, R_1 \right) \right\} \right), \\ R &= \begin{cases} P_{Cas} \left(\left\{ P_{Ohm} \left(\left\{ I, U_2 \right\}, R_1 \right) \right\}, V = \{I, R_1, U_2\}, \right. \\ \left. P_{Ohm} \left(\left\{ P_{Ohm} (U_2, R_2), U \right\}, V = \{R_1, R_2, U_2\}, \right. \end{cases} \quad (5) \\ U &= P_{Ohm} \left(\left\{ P_{Cas} \left(\left\{ P_{Ohm} (U_2, I), R_1 \right\}, I \right) \right\} \right), \\ I &= \begin{cases} P_{Ohm} \left(\left\{ P_{Cas} \left(\left\{ R, R_1 \right\}, U_2 \right) \right\}, V = \{R, R_1, U_2\}, \right. \\ \left. P_{Ohm} \left(\left\{ P_{Cas} \left(\left\{ R_1, R_2 \right\}, U \right) \right\}, V = \{R_1, R_2, U\}. \right. \end{cases} \end{aligned}$$

Для каждого из 8 полученных вариантов известные значения ДЗ можно также генерировать в допустимых диапазонах. Это позволяет на основании одной модели знаний предметной области задания получить большое множество ее вариантов, что значительно экономит время преподавателя на составление задач. При введении в модель дополнительного ДЗ U_2 количество вариантов структуры задания возрастает еще больше. Мы специально не стали вводить это ДЗ для сокращения математических выкладок.

3. Программная реализация инструмента моделирования знаний предметной области для задач по схемотехнике

Для того, чтобы предоставить разработчикам компьютерных обучающих программ возможность использовать модели знаний, сформированные специалистами в предметной области, необходимо обеспечить последних соответствующими компьютерными инструментами. Авторами статьи было разработано приложение, позволяющее в удобной графической форме редактировать модель знаний предметной области. Внешний вид окна приложения показан на рис. 3.

В основной части окна приложения отображается графический редактор модели задания, содержащий множества блоков ДЗ и ПЗ, а также взаимосвязей между ними. Эта структура дублируется в иерархическом дереве в левой панели. Для ДЗ имеется возможность добавления какого угодно количества слотов, содержащих характеристики. Например, для блока сопротивление (Resistance) можно добавить слоты «графическое обозначение», «определение», «единица измерения», «величина» и т.д. При этом каждому слоту должен соответствовать свой тип данных. В настоящее время поддерживаются четыре типа: String (строковый), Number (числовой), Image (графический), Boolean (логический). Для первых трех существует возможность использовать подстановочные значения (например, для случая, когда в подсказке, предлагаемой педагогической моделью, необходимо выбрать правильный вариант из перечня возможных или использовать в условии изображения), а для типа Number – возможность указывать диапазоны значений генерации и условия, накладываемые на генерируемое значение (например, для случая, когда задание с одним условием для каждого студента должно быть сгенерировано с различными значениями исходных данных). Для блоков ПЗ возможность настройки отсутствует. Вместо этого предлагается выбор требуемого ПЗ из заранее сформированной базы данных. После формирования модели знаний, компьютерное средство позволяет сгенерировать набор задач с различными значениями слотов типа Number, для которых были указаны диапазоны генерации. При этом существует возможность использовать различные искомые величины.

На основании анализа модели приложение автоматически определяет те исходные данные, которые необходимы для решения задачи по нахождению искомой величины или отображает сообщение о том, что это невозможно. Результатом генерации задачи является документ в формате xml.

Документ xml сам по себе не может использоваться в целях обучения. Вместо этого он должен

быть использован при формировании стратегии обучения педагогической моделью конкретной компьютерной обучающей программы.

Следует отметить, что модель, представленная на рис.3, не охватывает всех возможных способов определения входящих в нее декларативных знаний. Например, при нахождении величины R_1 при известных значениях U , R_2 и U_2 , студент мог бы использовать ПЗ, которое связывает напряжение источника с падением напряжения на каждом из резисторов R_1 и R_2 : $U=U_1+U_2$. Соответственно, в модели также не учтено падение напряжения на резисторе R_1 , U_1 . Полнота составляемой модели напрямую влияет на эффективность последующей работы педагогической модели. Модель на рис.3 была упрощена для наглядности. Это не значит, что педагогическая модель, работающая на основании этой модели, не будет вы-

полнять своей функции. Это означает, что студенту будет предложен единственно существующий алгоритм решения задачи (нахождение тока $I=U_2/R_2$, затем – нахождение общего сопротивления резисторов $R=U/I$, и в конце – определение сопротивления R_1 как $R_1=R-R_2$) для представленной модели. С другой стороны, рассмотренный недостаток может быть преодолен самой обучающей программой. Если студент использует решение, не охваченное моделью предметной области задания, но получает результат, совпадающий с результатом решения, которое можно сгенерировать на основании ДЗ и ПЗ модели, то в обучающей программе должен быть предусмотрен механизм идентификации и регистрации студенческого решения. Рекомендованная архитектура педагогической модели, поддерживающей регистрацию студенческих решений, приведена на рис. 4.



Рис. 4. Структура педагогической модели, рекомендованная для анализа моделей заданий

Определение верности ответа и необходимость подсказки в такой архитектуре основывается на методе сравнения абстрактных синтаксических деревьев, построенных в соответствии со студенческой и эталонной моделью решения. Подобная архитектура уже применялась авторами ранее, в приложении SQLTOR, компьютерной программе, обучающей языку запросов к базам данных SQL [8].

Результаты

В результате исследования была предложена форма представления знаний, при помощи которой удобно моделировать предметную область заданий, с точки зрения дальнейшего использования ее педагогической моделью обучающих программ. Как было показано выше, такие модели представления знаний

обеспечивают возможность генерации обучающих заданий, как с различными значениями известных величин, так и с различными их наборами (как следствие – с различными условиями заданий). Для того, чтобы обеспечить возможность создания моделей специалистами в предметной области, не знакомыми или слабо знакомыми с программированием, было разработано компьютерное приложение, позволяющее создавать модели в графическом редакторе. Для обеспечения программистов компьютерных обучающих программ удобным интерфейсом использования создаваемых моделей предусмотрен экспорт в формат xml.

Перспективной задачей является разработка компьютерной обучающей программы, педагогическая модель которой будет полностью поддерживать модели представления знаний и соответствовать

архитектуре, рассмотренным в статье. Также планируется дальнейшее исследование моделей представления знаний для различных классов задач по схемотехнике с целью определения области их возможного применения.

Литература

1. VanLehn, K. *The behavior of tutoring Systems [Text]* / K. VanLehn // *International Journal of Artificial Intelligence In Education*. – Athens (Greece). – 2006. – №16(3). – P. 227 – 265.

2. *Opening the Door to Non-programmers: Authoring Intelligent Tutor Behavior by Demonstration [Text]* / K.R. Koedinger, V. Aleven, N. Heffernan, B. McLaren, M. Hockenberry // *7th Int. Conf. Intelligent Tutoring Systems*. – Springer, Maceio (Brazil). – 2004. – P. 162 – 174.

3. Ainsworth, S. *Teachers implementing pedagogy through REDEEM [Text]* / S. Ainsworth, S. Grimshaw, J. Underwood // *Computers & Education*. – 1999. – V. 33, N. 2-3. – P. 171 – 187.

4. Martin, B. *ITS Domain Modelling with Ontology [Text]* / B. Martin, A. Mitrovic, P. Suraweera // *Journal of Universal Computer Science*. – 2008. – V. 14, O. 17. – P. 2758 – 2776.

5. Sleeman, D. *Intelligent Tutoring Systems [Text]* / D. Sleeman, J.S. Brown // *Academic Press*. – 1982. – P. 227 – 282.

6. Gertner, A. *Procedural help in Andes: Generating hints using a Bayesian network student model [Text]* / A. Gertner, C. Conati, K. VanLehn // *Proceedings of the Fifteenth National Conference on Artificial Intelligence AAAI-98*. – 1998. – P. 106 – 111.

7. *Building intelligent interactive tutors [Text]: Student-centered strategies for revolutionizing e-learning*. – Beverly Park Woolf. Department of Computer Science, University of Massachusetts. – Amherst (USA). – 2008. – 480 p.

8. Чухрай, А.Г. *Метод компьютерного обучения SQL с помощью абстрактных синтаксических деревьев [Текст]* / А.Г. Чухрай, З.В. Томченко, Е.С. Вагин // *Системы управління, навігації та зв'язку*. – 2011. – Вип. 2 (18). – С. 232 – 237.

Поступила в редакцію 7.02.2012

Рецензент: д-р техн. наук, проф., зав. кафедрой 303 Н.Д. Кошевой, Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», Харьков, Украина.

МОДЕЛІ ПРЕДСТАВЛЕННЯ ЗНАНЬ В ІНТЕЛЕКТУАЛЬНИХ КОМП'ЮТЕРНИХ ПРОГРАМАХ, ЩО НАВЧАЮТЬ СХЕМОТЕХНІЧНИМ РІШЕННЯМ

А.Г. Чухрай, Є.С. Вагин, Т.І. Шевчук

Зроблено огляд різних підходів до представлення знань в існуючих комп'ютерних системах навчання. Запропоновано авторський підхід до представлення знань предметної області завдань зі схемотехніки, що базується на розділенні знань на декларативні та процедуральні, а також дозволяє групувати їх в ієрархічні дерева компонентів компетенції. На прикладі розглянуто процес генерації набору варіантів завдання, що використовує запропоновану модель представлення знань в області схемотехніки. Описано авторське програмне забезпечення, що полегшує процес створення подібних моделей.

Ключові слова: комп'ютерні навчальні програми, алгоритмізація навчання, компонент компетенції.

KNOWLEDGE REPRESENTATION MODELS IN INTELLECTUAL COMPUTER PROGRAMS FOR CIRCUIT ENGINEERING SOLUTIONS TUTORING

A.G. Chukhray, Ie.S. Vagin, T.I. Shevchuk

Short review of different approaches for knowledge representation in existing computer tutoring systems is made. Author's approach of knowledge representation for tasks in circuit engineering domain is proposed. This approach is based on representation of knowledge as declarative and procedural elements, which allows grouping of these knowledge elements into hierarchical trees of competence components. Example of generation of task variants set based on proposed knowledge representation model in circuit engineering field is considered. Author's software for such domain knowledge modeling is described.

Keywords: computer tutoring programs, tutoring algorithmization, competence component.

Чухрай Андрей Григорьевич – канд. техн. наук, доцент, докторант кафедры систем управления летательных аппаратов Национального аэрокосмического университета им. Н.Е. Жуковского «ХАИ», Харьков, Украина, e-mail: achukhray@gmail.com.

Вагин Евгений Сергеевич – аспирант кафедры систем управления летательных аппаратов Национального аэрокосмического университета им. Н.Е. Жуковского «ХАИ», Харьков, Украина, e-mail: ie.s.vagin@gmail.com.

Шевчук Татьяна Ивановна – ведущий специалист кафедры систем управления летательных аппаратов Национального аэрокосмического университета им. Н.Е. Жуковского «ХАИ», Харьков, Украина, e-mail: t.i.shevchuk@gmail.com.