

УДК 681.3.07

А.С. КУЛИК¹, А.Г. ЧУХРАЙ¹, П. АНЦЕНБЕРГЕР^{1,2}¹Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», Украина²Университет прикладных наук, Вельс, Австрия

ЗАДАЧИ РАЗРАБОТКИ АДАПТИВНОГО КОМПЬЮТЕРНОГО СРЕДСТВА ОБУЧЕНИЯ SQL

Приведен анализ существующих адаптивных компьютерных средств обучения профессиональным умениям в разных предметных областях с точки зрения воплощенных в них функций и педагогических принципов. Описаны современные разработки, обучающие умения формировать запросы на языке Structured Query Language (SQL), что является одним из наиболее общих и важных требований предъявляемых сегодня к специалистам в области информационных технологий. Представлены актуальные задачи разработки адаптивного компьютерного средства обучения SQL.

Ключевые слова: адаптивное компьютерное средство обучения, SQL, внешний цикл, внутренний цикл.

Введение

В последнее время одними из наиболее востребованных на мировом рынке труда специальностей стали IT-специальности. От качества подготовки IT-специалистов напрямую зависит качество, в том числе и надежность, различных компьютерных систем, к услугам которых ежедневно обращаются миллионы людей.

Вместе с тем традиционное обучение в условиях массового производства даже при наличии очень хорошего педагога обладает существенным недостатком – принципиальной невозможностью адаптироваться к каждому обучаемому.

С 60-х гг. XX столетия исследователи во всем мире разрабатывают различные компьютерные средства обучения (КСО), обладающие, в той или иной степени, адаптивными функциями. Тем не менее, задача разработки совершенного компьютерного педагога еще далека от своего разрешения. В статье представлены актуальные задачи разработки адаптивного КСО (АКСО) умениям формировать SQL-запросы, что является одним из наиболее общих и важных требований предъявляемых сегодня к IT-специалистам. Для этого приведен анализ существующих АКСО с точки зрения воплощенных в них перспективных адаптивных функций и современных педагогических принципов.

1. Современные функции АКСО

Как показано в широко цитируемой обзорной работе [1] многие АКСО могут быть представлены, как состоящие из двух циклов: внешнего и внутреннего.

Внешний цикл ответственен за принятие решения по выбору следующей задачи обучаемому. Внутренний цикл предназначен для пошаговой помощи при решении задачи.

Внешний цикл. Рассмотрим, как реализуют возможность выбора следующей задачи существующие АКСО.

Формирование множества задач для последующего выбора. Чаще всего задачи составляются педагогами, при этом разработчики создают оболочки для облегчения занесения задач в систему [2].

Вместе с этим ряд компьютерных систем самостоятельно решают задачи [1].

Например, инструментальное средство для известной системы Andes получает на входе формальное описание физической задачи, а также текст и графику. После этого данное средство пытается решить задачу самостоятельно. Если не выходит, тогда оно просит помощи у автора задачи [3].

Для некоторых доменов задач средства не могут решить задачу, но когда наблюдают решение педагога, способны обобщить это решение. Например, система Steve ожидает, что педагог введет длинную последовательность виртуальных действий (открытие клапанов, нажатие кнопок, ...).

Система обобщает последовательность в процедуру для достижения заданной цели. Для того чтобы обобщить, система задает педагогу ряд вопросов [4].

Тем не менее, существуют и генераторы задач. Например, на кафедре систем управления летательных аппаратов Национального аэрокосмического университета им. Н.Е. Жуковского «ХАИ» создана web-система для подготовки выпускников общеоб-

разовательных учебных заведений к внешнему независимому оцениванию по математике [5]. Особенностью системы является наличие генераторов 26 типов математических задач, включая алгебраические и геометрические задачи. Вместе со случайной генерацией исходных данных для постановок задач генерируются и правильные, а в случае закрытых тестов и неправильные, но правдоподобные решения.

Аналогично для заданной спецификации компонента умений, который необходимо изучить, в системе SQLTutor генерируется запрос на языке SQL. После этого педагог формулирует на естественном языке (в данном случае – английском) постановку задачи. Таким образом, за 3 часа педагог в состоянии создать 200 задач, достаточных для шестинедельного модуля по курсу SQL [6].

Выбор задачи. В работе [1] для выбора следующей задачи описывается шесть основных подходов, используемые в существующих разработках. Эти подходы упорядочены по возрастанию сложности их реализации.

1) система отображает меню и предоставляет обучаемому возможность выбора следующей задачи. Например, такая возможность реализована в системе Andes [3].

2) система формулирует задачи в фиксированной последовательности. Такой подход реализован во многих АКСО, в том числе AutoTutor [7], «Обучающие программы по дисциплине «Моделирование систем» [8] и других. Главное преимущество в том, что система побуждает овладевать умениями.

3) система реализует педагогический принцип «достижение мастерства». Система выбирает задачи из текущего уровня сложности до тех пор, пока студент не овладеет им. Принцип «достижение мастерства» требует наличия средств оценивания компетентности студентов на текущем уровне сложности. Такой подход реализует, в частности, система Sherlock [9].

4) система реализует педагогический принцип «макроадаптация». Например, это делает Algebra Cognitive Tutor [10]. Система знает, из каких компонентов компетентности состоит каждая задача и может предложить задачу, которая требует несколько компонентов компетентности уже освоенных студентом и двух не освоенных компонентов. В дополнение к представлению компонентов компетентности, в некоторых системах представлены также компоненты некомпетентности.

Ряд систем учитывает также стили изучения и предпочтения, например, предпочтение визуального или вербального объяснения.

В дополнение к выбору задачи некоторые сис-

темы предлагают выбор режима задачи. Например, система Steve в первом режиме демонстрирует решение задачи по шагам с объяснениями, во втором режиме – подсказывает каждый шаг, перед тем как его выполнит студент, а в третьем режиме предлагает студенту решить задачу без предоставления подсказок. Для выбора режима, как правило, используется такой педагогический принцип как «постепенное отвыкание от подпорок» [1].

5) некоторые АКСО могут начать работу со студентом, например, с прочтения нескольких страниц гипертекста. Следующая задача может заключаться в просмотре видео-фрагмента по проблеме. Затем несколько задач могут быть предложены для самостоятельного решения студентом.

Наконец, АКСО может потребовать от студента пройти тест. Поскольку сложные дисциплины почти всегда характеризуются иерархической структурой подачи материала, то и выбор задач осуществляется на каждом уровне иерархии. В данном случае можно говорить о пятом, многоуровневом подходе к выбору следующей задачи, в котором на каждом уровне может применяться один из четырех описанных выше подхода.

6) внешний цикл может реализовывать и более сложные технологии выбора [11]. Например, АКСО может спланировать определенную последовательность задач для конкретного студента базируясь на модели компетентности этого студента, и затем начать выполнять эту последовательность. В середине последовательности АКСО может обнаружить, что у студента серьезные пробелы в умениях и перестроить план.

Внутренний цикл. В то время как внешний цикл необходим для выбора следующей задачи, внутренний цикл работает в контексте решаемой в данный момент задачи и предназначен для оказания пошаговой помощи в решении.

Как показано в обзоре [1] предоставляемую различными системами помощь можно обобщить и представить в виде таких наиболее распространенных сервисов как:

- 1) минимальная пошаговая диагностическая информация;
- 2) специфическая диагностическая информация в случае неправильного шага;
- 3) подсказка следующего шага;
- 4) оценивание умений;
- 5) обзор решения.

Рассмотрим более подробно пять вышеуказанных сервисов внутреннего цикла. Для доступности дальнейшего изложения будем рассматривать внутренний цикл АКСО, как состоящий из двух основных функций: генерации шага и анализа шага.

1) минимальная пошаговая диагностическая информация.

Категории минимальной диагностической информации. Большинство систем используют только две категории: правильно и неправильно. Вместе с тем, как показывает практика, существуют и другие категории.

Например, текущий шаг может не быть частью идеального решения, а являться частью неидеального, но правильного решения. В таком случае система может реагировать третьим способом: правильно, но не оптимально. Такая тактика была реализована в системе Sherlock. В том случае, если шаг не может быть отнесен в одну из категорий (правильно, неправильно, не оптимально), он может быть зарегистрирован, как не распознанный.

Существует три основных подхода к не распознанным шагам: 1) в некоторых системах, таких как Algebra Cognitive Tutor заложено предположение, что все правильные шаги могут быть распознаны. Следовательно, все не распознанные шаги определяются, как неправильные; 2) в других системах, таких как SQL-Tutor действует предположение, что студенты могут иногда синтезировать новые, правильные решения, т.е. не распознанные шаги воспринимаются как правильные; 3) АКСО может выдавать сообщение о том, что шаг не распознан.

Разделение по времени реакции системы

Существуют также различные тактики предоставления диагностической информации: а) немедленная диагностическая информация (системы Andes, Algebra Cognitive Tutor, Steve, AutoTutor); б) диагностическая информация с задержкой (Sherlock); в) диагностическая информация по требованию (SQL-Tutor).

В дополнение к перечисленным тактикам существуют и более сложные. АКСО может после каждого шага принимать решения о том, когда предоставлять диагностическую информацию. Тактика предоставления диагностической информации может быть функцией мастерства студента.

Например, если студент почти достиг уровня компетентности, диагностическая информация может быть с задержкой, в то время как для этой же задачи, но для другого, менее компетентного, студента диагностическая информация может быть немедленной, т.е. и здесь можно исходить из принципа «постепенное отвыкание от подпорок».

3) Подсказка следующего шага;

АКСО, которая предоставляет минимальную диагностическую информацию, должна также иметь возможность подсказать следующий шаг. Без минимальной диагностической информации студенты часто пребывают в неведении о своих ошибках весь

путь к неправильному решению. С минимальной диагностической информацией, но без подсказки следующего шага они могут долго угадывать и затем прийти в состояние фрустрации. Таким образом, подсказка следующего шага должна быть общим сервисом АКСО, обеспечивающих диагностическую информацию.

Когда подсказывать. Уместный здесь педагогический принцип заключается в том, что АКСО должна подсказывать следующий шаг только тогда, когда студент действительно нуждается в этом. Если студент самостоятельно в результате продуктивной умственной работы может найти правильный шаг, система может не подсказывать. С другой стороны, если студент впал в состояние фрустрации или многократно и напрасно пытается угадать следующий правильный шаг, АКСО должна предоставить подсказку, даже если ее не просят об этом. Такая тактика реализована в системе DT Tutor [12].

Более простые тактики используют системы Andes, Sherlock и другие. В некоторых системах подсказка выдается по требованию студента. Тем не менее, наличие кнопки «Подсказка» характеризуется двумя известными проблемами [13]:

а) избыточное использование – некоторые студенты запрашивают подсказку даже тогда, когда они не нуждаются в ней. В экстремальных случаях подсказка запрашивается на каждом шаге;

б) недоиспользование – некоторые студенты отказываются обращаться к подсказкам даже тогда, когда они действительно нуждаются в этом.

Для того чтобы препятствовать избыточному использованию подсказок, некоторые системы ведут счет и уменьшают баллы за каждую подсказку. С другой стороны, для предотвращения недоиспользования подсказок некоторые АКСО предоставляют подсказки после трех попыток ввести правильный шаг или в случае длительной паузы.

Существует и другая тактика: проактивного поведения АКСО, т.е. выдачи подсказок перед попыткой сделать шаг. Например, если DT Tutor оценивает, что следующий шаг студента без подсказки будет ошибочным, то система говорит нечто подобное следующему “Следующий шаг – хитрый. Помни, что правило цепочки $dx/dz = dx/dy * dy/dz$.” АКСО Steve использует еще более экстремальную форму проактивной подсказки. Если следующий шаг ранее не делался или не наблюдался студентом, то система говорит «Следующий шаг – новый. Позволь мне показать, как это делается» и выполняет следующий шаг самостоятельно.

Какие шаги подсказывать.

Критериями выбора шагов для генерации могут быть следующие:

- а) шаг должен быть правильным;
- б) подсказываемый шаг должен быть новым для студента;
- в) должны быть учтены предпочтения преподавателя;
- г) если студент имеет план решения задачи, АКСО должна подсказать шаг, необходимый для завершения решения.

Как подсказывать следующий шаг. Наиболее общий педагогический принцип – заставлять студентов как можно больше думать самостоятельно, т.е. первая подсказка должна быть минимальна.

Еще один используемый принцип – если более чем один компонент компетентности включен в подсказку, то весьма трудно сформировать педагогически эффективную последовательность подсказок. Последовательность сложных подсказок может быть хуже, чем ее отсутствие. В таком случае лучше дать сразу самую детальную подсказку.

Например, в работе [14] используется последовательность из трех типов подсказок: указать, научить, детализировать.

Последовательности подсказок могут иногда быть упрощены путем отслеживания ранее выданных. Достаточно простой случай возникает тогда, когда АКСО и студент обсудили цель и план и эти цель или план все еще актуальны.

4) специфическая диагностическая информация в случае неправильного шага.

Существует множество методов для диагностирования ошибок. Большинство из них требует, чтобы авторы наблюдали студенческие ошибки, вычисляли, что же вызвало наиболее общие из них, описывали соответствующие типы ошибок и реализовали их обнаружение.

Когда предоставлять специфическую диагностическую информацию

Рассмотрим существующие тактики в этом вопросе.

АКСО Algebra Cognitive Tutor предоставляет минимальную диагностическую информацию, если только студент безрезультатно пытался выполнить один и тот же шаг дважды.

Тогда система предоставляет первую подсказку из последовательности подсказок для этого шага. Если студент вводит шаг с известной системе ошибкой, тогда уже предоставляются специфические подсказки перед подсказками следующего шага.

Реализованная в системе Andes тактика – обеспечение немедленной минимальной диагностической информации по требованию (при нажатии на кнопки). Существует два вида кнопок: обеспечивающие подсказки следующего шага и обеспечивающие диагностирование текущего шага.

Иногда студенты делают ошибки по невнимательности [15], но не замечают их, что приводит к потере времени на глубокий поиск концептуальных ошибок. Andes разделяет типы ошибок на концептуальные и ошибки по невнимательности. Ошибки по невнимательности часто допускаются в том числе экспертами, включая преподавателей.

5) оценивание. Как отмечают, в частности, авторы публикаций [1, 16] легко увлечься этой частью и сделать ее перегруженной. В то же время эта часть не является основной частью АКСО.

6) обзор решения. Множество АКСО предоставляют диагностическую информацию и подсказки только после того, как студент закончит решение. Например, так функционирует DC-Train [17], созданная с целью обучения морских офицеров борьбе с пожарами, затоплениями и другими катастрофами. Даже в предметных областях не реального времени диагностическая информация с задержкой более эффективна с точки зрения развития метакогнитивных знаний и умений.

Далее рассмотрим особенности существующих АКСО в выбранной предметной области – области баз данных.

2. Анализ существующих компьютерных средств обучения SQL

Наиболее известной АКСО SQL является система SQL-Tutor [18, 19], разработанная в университете Canterbury, Новая Зеландия под руководством A. Mitrovic. Каждая задача в ней состоит из модели данных и постановки, какую информацию нужно получить. Студенты пишут SQL-запросы, заполняя бланки.

После завершения работы над запросом они могут нажать кнопку «Представить решение на рассмотрение». АКСО анализирует студенческий запрос с целью нахождения ошибок и предоставляет множество уровней диагностической информации и подсказок. Студенты самостоятельно выбирают уровень диагностической информации, которую желают получить. После получения диагностической информации и редактирования запроса студент снова может переслать его на рассмотрение системой.

Для анализа шагов SQL-Tutor содержит множество ограничений, каждое из которых включает в себя две части: условие релевантности и условие удовлетворения ограничения [20]. Если условие релевантности не выполняется, то АКСО никак не реагирует. Если же оно выполняется и выполняется условие удовлетворения ограничения, то АКСО снова никак не реагирует. Реакция происходит толь-

ко в том случае, если первое условие выполняется, а второе – нет. Это значит, что ограничение нарушено и АКСО идентифицирует соответствующие обратные связи. Детально, каждое ограничение имеет два сообщения. В зависимости от уровня диагностической информации, выбранного АКСО либо студентом, первое, включающее короткое описание нарушения ограничения, может быть предоставлено студенту, когда ограничение нарушено. Второе – более детализированное.

К недостаткам разработки можно отнести то, любое ограничение может быть сведено к давно известным правилам «если-то». Это утверждается, в частности, и в исследовании [21]. Вместе с тем в SQL-Tutor содержатся недопустимые ошибки. Например, ограничение 476 базируется на неверном предположении о зависимости между количеством таблиц в разделе FROM и количеством условий в разделе WHERE.

В то же время существует множество практических задач, в которых требуется нахождение только лишь декартова произведения отношений. Приведены также примеры ошибок системы первого и второго рода. Ряд ограничений явно создан с целью синтаксического сравнения между единственным идеальным решением и решением студента. Естественно, что эти ограничения не дают шансов альтернативным решениям, которые могут быть более рациональными, чем созданные авторами.

Альтернативный и часто противопоставляемый подход, реализующийся в виде правил-моделей ошибок представлен в работе [22].

В ней описаны 37 классов наиболее распространенных, по мнению авторов, семантических ошибок, допускаемых студентами при составлении SQL-запросов. Для обнаружения описанных классов ошибок авторами реализована компьютерная система sqlint. Очевидный недостаток – отсутствие доказательств полноты множества классов ошибок.

Также следует выделить такие системы, покрывающие только часть умений, как eSQL [23] – для обучения концепции обработки запросов; AsseSQL [24] и SQLator [25] – для оценивания запросов студентов, SQLify [26] – средство, реализующее семантическую обратную связь, улучшенное автоматическое оценивание, а также предоставляющее возможность студентам оценивать решения друг друга.

Вместе с тем задача создания более совершенной АКСО в области баз данных, как впрочем и в любой другой, далека от своего разрешения. Перспективно двигаться вперед в плане реализации адаптивных функций и педагогических принципов, частично описанных выше. Для выбранной пред-

метной области актуальными научными задачами являются следующие.

3. Актуальные научные задачи

1. Разработать модели АКСО умениям составлять SQL-запросы путем решения задач трех уровней: задач реляционной алгебры и реляционного исчисления, задач с использованием стандарта SQL и задач с использованием SQL промышленной системы управления базами данных.

2. Разработать метод адаптивного выбора следующей задачи для конкретного студента, отличающийся от известных спецификой предметной области и учетом современных педагогических принципов.

3. Сформировать метод адаптивной гибкой помощи при решении текущей задачи, отличающийся от известных спецификой предметной области и учетом современных педагогических принципов.

4. На основе решений первых трех задач разработать АКСО SQL и провести его экспериментальные исследования и доработку.

Заключение

На основе анализа существующих адаптивных компьютерных средств обучения профессиональным умениям в разных предметных областях с точки зрения воплощенных в них функций и педагогических принципов, а также анализа известных систем обучения языку SQL сформулированы актуальные научные задачи разработки адаптивного компьютерного средства обучения SQL.

Литература

1. VanLehn K. *The Behavior of Tutoring Systems* / K. VanLehn // *International Journal of Artificial Intelligence in Education*. – 2006. – Vol. 16. – P. 227-265.
2. Murray T. *Authoring Intelligent Computer Systems: An analysis of the state of the art* / T. Murray // *International Journal of Artificial Intelligence in Education*. – 1999. – P. 98-129.
3. VanLehn K. *The Andes physics tutoring system: Lessons learned* / K. VanLehn, C. Lynch, K. Schultz, J.A. Shapiro, R.H. Shelby, L. Taylor // *International Journal of Artificial Intelligence in Education*. – 2005. – № 15(3). – P. 147-204.
4. Johnson W.L. *Integrating pedagogical agents into virtual environments* / W.L. Johnson, J. Rickel, R. Stiles, A. Munro // *Presence: Teleoperators and Virtual Environments*. – 1998. – № 7(6). – P. 523-546.

5. Kulik A. *Models and software for intelligent web-based testing system in mathematics* / A. Kulik, A. Chukhray, S Rakov, S. Pedan // *In Proceedings of the East-West Fuzzy Colloquium, Zittau, Germany.* – 2008. – P. 63-72.
6. Martin B. *Automatic problem generation in constraint-based tutors* / B. Martin, A. Mitrovic // *Intelligent Tutoring Systems: 6th International Conference, Berlin: Springer ITS, 2002.* – P. 388-398.
7. Graesser A.C. *AutoTutor: A tutor with dialogue in natural language* / A.C. Graesser, S. Lu, G.T. Jackson, H.H. Mitchell, M. Ventura, A. Olney // *Behavioral Research Methods, Instruments and Computers.* – 2004. – № 36. – P. 180 - 193.
8. Kulik A. *Development of the automated laboratory practical work at the course «Modeling of systems»* / A. Kulik, A. Chukhray, S. Pedan, P. Anzenberger // *Proc. of the Interactive Computer Aided Learning Conference, Villach, Austria, Sept. 24 – 26 2008.* – P. 10.
9. Katz S. *Sherlock 2: An intelligent tutoring system built on the LRDC framework* / S. Katz, A. Lesgold, E. Hughes, D. Peters, G. Eggan, M. Gordin // *Facilitating the development and use of interactive learning environments.* – 1998. – P. 227-258.
10. Anderson J.R. *Cognitive Tutors: Lessons Learned* / J.R. Anderson, A.T. Corbett, K.R. Koedinger, R. Pelletier // *Journal of the Learning Sciences.* – 1995. – № 4(2). – P. 167-207.
11. Zinn C. *A 3-tier planning architecture for managing tutorial dialogue* / C. Zinn, J.D. Moore, M.G. Core // *Proceedings of the Intelligent Tutoring Systems Sixth International Conference 2002.* – P. 574-584.
12. Murray C. *Looking ahead to select tutorial actions: A decision-theoretic approach* / C. Murray, K. VanLehn, J. Mostow // *International Journal of Artificial Intelligence in Education.* – 2004. – № 14(3-4). – P. 235-278.
13. Alevin V. *Limitations of student control: Do students know when they need help?* / V. Alevin, K.R. Koedinger // *Intelligent Tutoring Systems: 5th International Conference, ITS, 2000.* – P. 292-303.
14. Hume G. *Hinting as a tactic in one-on-one tutoring* / G. Hume, J. Michael, A. Rovick, M. Evens // *Journal of the Learning Sciences.* – 1996. – № 5(1). – P. 23-49.
15. Kulik A. *Diagnostic models of intelligent tutor system for teaching skills to solve algebraic equations* / A. Kulik, A. Chukhray, M. Chukhray // *In Proc. of the Interactive Computer Aided Learning Conference, Villach, Austria, Sept. 26–28 2006, International Journal of Emerging Technologies in Learning.* – 2006. – Vol. 2. – № 1. – P. 27-35.
16. Self J.A. *Bypassing the intractable problem of student modeling* / J.A. Self // *Intelligent Tutoring Systems: At the crossroads of AI and Education, 1990.* – P. 107-123.
17. Bulitko V.V. *Automated instructor assistant for ship damage control* / V.V. Bulitko, D.C. Wilkins // *In Proc. of AAAI-99, 1999.* – Vol. 1. – № 3. – P. 57-66.
18. Mitrovic A. *Learning SQL with a computerised tutor* / A. Mitrovic // *29th ACM SIGCSE Technical Symposium, Atlanta, 1998.* – P. 307-311.
19. Mitrovic A. *An intelligent SQL tutor on the web* / A. Mitrovic // *International Journal of Artificial Intelligence in Education 13, 2003.* – P. 171-195.
20. Ohlsson S. *Constraint-based Student Modeling* / S. Ohlsson // *Proc. Student Modelling: the Key to Individualized Knowledge-based Instruction, Springer-Verlag, 1994.* – P. 167-189.
21. Viswanathan K. *An Assessment of Constraint-Based Tutors: A Response to Mitrovic and Ohlsson's Critique of "A Comparison of Model-Tracing and Constraint-Based Intelligent Tutoring Paradigms"* / K. Viswanathan, R. Weitz, D. Rosenthal // *International Journal of Artificial Intelligence in Education.* – 2006. – Vol. 16. – P. 291-321.
22. Brass St. *Detecting Logical Errors in SQL Queries* / St. Brass, Chr. Goldberg // *Technical Report, University of Halle, 2004.*
23. Kearns R. *A teaching system for SQL* / R. Kearns, S. Shead, A. Fekete // *In Proceedings of ACSE '97, Melbourne, 1997.* – P. 224-231.
24. Prior J. *The backwash effect on SQL skills grading* / J. Prior, R. Lister // *In Proceedings of ITiCSE'04, Leeds, UK, 2004.* – P. 32-36.
25. Sadiq Shazia, Orlowska Maria, Sadiq Wasim, Lin Joe *SQLator—an online SQL learning workbench* /
26. Sadiq S. / S. Sadiq, M. Orlowska, W. Sadiq, J. Lin // *Proc. of ITiCSE'04, Leeds, UK, 2004.* – P. 223-227.
27. Dekeyser S. *Computer Assisted Assessment of SQL Query Skills* / S. Dekeyser, // *Proc. of the 18 conf. on Australasian database.* – 2007. – Vol. 63. – P. 53 - 62.

Поступила в редакцію 2.02.2009

Рецензент: д-р пед. наук, проф., зам. директора центра С.А. Раков, Український центр оцінювання якості освіти, Україна.

ЗАДАЧІ РОЗРОБКИ АДАПТИВНОГО КОМП'ЮТЕРНОГО ЗАСОБУ НАВЧАННЯ SQL*А.С. Кулік, А.Г. Чухрай, П. Анценбергер*

Наведено аналіз існуючих адаптивних комп'ютерних засобів навчання професійним умінням у різних предметних областях з погляду втілених у них функцій і педагогічних принципів. Описано сучасні розробки, що навчають умінням формувати запити мовою Structured Query Language (SQL), що є однією з найбільш загальних і важливих сучасних вимог до фахівців в області інформаційних технологій. Представлено актуальні задачі розробки адаптивного комп'ютерного засобу навчання SQL.

Ключові слова: адаптивний комп'ютерний засіб навчання, SQL, зовнішній цикл, внутрішній цикл.

THE TASKS OF DEVELOPMENT OF ADAPTIVE COMPUTER TOOL FOR SQL TUTORING*A.S. Kulik, A.G. Chukhray, P. Anzenberger*

The analysis of existed adaptive computer tools for professional skills tutoring in different domains is presented. The tools are described from implemented functions and pedagogic principles the point of view. The modern developments for tutoring of skills to form Structured Query Language (SQL) queries are described because these skills are the most common and important requirements to IT-specialist. Up-to-date research tasks of development of adaptive computer tool for SQL tutoring are composed.

Key words: adaptive computer tool for SQL tutoring, external loop, internal loop.

Кулик Анатолій Степанович – д-р техн. наук, проф., декан факультета систем управління летательных аппаратов, зав. кафедрой систем управления летательных аппаратов Национального аэрокосмического университета им. Н.Е. Жуковского «ХАИ», Харьков, Украина, e-mail: kulik@d3.khai.edu.

Чухрай Андрей Григорьевич – к.т.н., заместитель декана факультета систем управления летательных аппаратов по НИР, доцент кафедры систем управления летательных аппаратов Национального аэрокосмического университета им. Н.Е. Жуковского «ХАИ», Харьков, Украина, e-mail: chukhray@d3.khai.edu;

Анценбергер Петер – аспирант кафедры систем управления летательных аппаратов Национального аэрокосмического университета им. Н.Е. Жуковского «ХАИ», преподаватель университета прикладных наук, Вельс, Австрия. e-mail: peter@peter-partner.com.