

УДК 378.16:681.3

В.В. Романенко

Использование средств проектирования данных для разработки компонентов компьютерного учебного курса

Рассказывается о способах применения автоматизированных средств визуального проектирования структур данных для разработки различных видов интерактивных компонентов в компьютерных учебных программах (тренажерах, блоках контроля и самоконтроля и т.д.).

Введение

Лаборатория «Мультимедиа», созданная на базе кафедры автоматизированных систем управления (АСУ) ТУСУРа и Томского межвузовского центра дистанционного образования (ТМЦ ДО), уже в течение нескольких лет занимается разработкой компьютерных учебных программ (КУП). За это время был создан ряд средств, автоматизирующих в той или иной степени разработку различных видов КУП, — гипермедийных электронных учебников, анимированных лекций, тренажеров, систем контроля знаний. Все они в совокупности образуют единый автоматизированный комплекс EduCAD [1].

Компоненты данного комплекса постоянно развиваются, и в настоящее время разрабатывается универсальный редактор, позволяющий проектировать структуру данных для различных видов КУП. Необходимость использования средств проектирования при разработке программного обеспечения (ПО) очевидна. В настоящее время развиваются такие мощные языки проектирования, как UML [2], а средства проектирования включены во многие пакеты разработки ПО (Borland Delphi/C++ Builder, MS Visual Studio). Однако средства проектирования можно применять не только при разработке ПО, но и при разработке структур данных. Это подтверждает использование таких средств в различных СУБД и т.д.

Каким же образом средства проектирования данных могут быть использованы при разработке КУП?

Использование средств проектирования для разработки КУП

Все автоматизированные средства комплекса EduCAD состоят из двух частей — редактора и клиента. Редактор позволяет создать учебную базу данных, а клиент — отобразить ее содержимое ученику. Таким образом, необходимо максимально упростить именно создание базы данных, т.к. в этом и заключается автоматизированная разработка КУП.

Рассмотрим структуру данных гипермедийного электронного учебника. Обычно он строится по фреймовой технологии обучения, базирующейся на формировании абстрактных образов (кадров) и отношений (связей) между ними [3]. В данном случае связями являются ссылки на различные учебные ресурсы и другие кадры учебника, а также переходы на другие кадры согласно навигационной модели. Дело осложняется тем, что навигационных моделей в одном и том же учебнике может быть несколько. Это нужно для того, чтобы по одному и тому же учебнику могли заниматься студенты различных специальностей. В этом случае учебник содержит все необходимые материалы, а траектория обучения зависит от навигационной модели. Более того, в некоторых учебниках каждый волен сам выбирать траекторию обучения.

Все это приводит к тому, что схема данных содержит множество блоков и еще больше связей между ними (рис. 1). Вручную редактировать эти связи при возрастании объема учебного материала становится затруднительно.

На рис. 1 рассмотрен всего один блок, но уже видна вся сложность проблемы. Если еще попытаться обеспечить возможность создания студентам своих траекторий обучения, то придется вводить новый вид «плавающих» связей. Кроме того, при большом количестве кадров охватить полную картину учебной информации становится невозможно [4]. Визуальное проектирование данных поможет:

- более наглядно обрабатывать связи, причем группами;
- использовать множество видов связей и блоков;

- создавать многоуровневые схемы. Так, например, несколько фреймов можно объединить в один блок (раздел, параграф) и далее работать с ним как единым целым, и так далее вверх по иерархии материала;
- визуально добавлять и удалять связи и блоки;
- импортировать и экспортировать данные в различные форматы других систем проектирования.

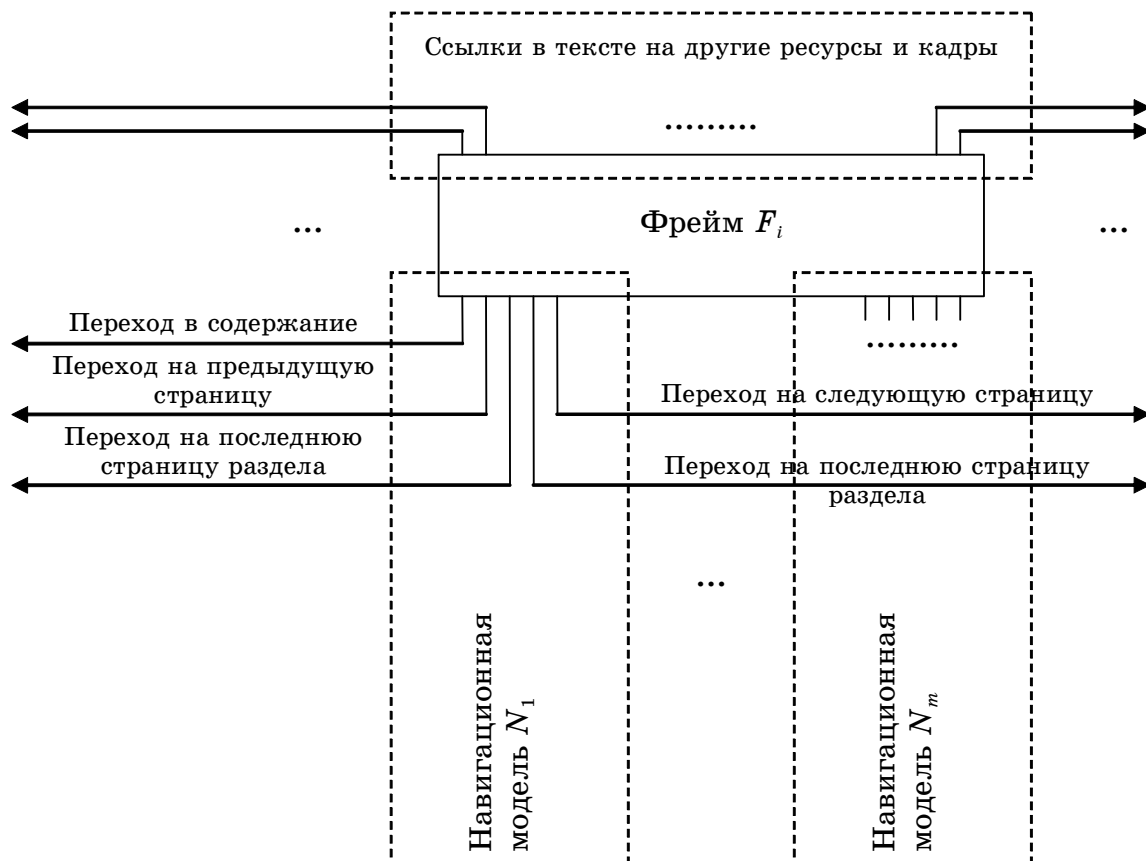


Рис. 1. Участок схемы данных гипермедийного учебника

То же самое можно сказать и относительно автоматизированных систем контроля знаний (САКЗ). Обычная база данных САКЗ не содержит никаких связей, т.к. вопросы из нее при тестировании выбираются случайным образом. Но если речь идет об адаптивном тестировании [5], то связи появляются, причем разных видов. Под адаптивным тестированием (АТ) понимается ситуация, когда вопрос, задаваемый студенту, зависит от результатов его предыдущих ответов. Обычно используются две схемы АТ:

1) с жесткими связями. В этом случае для каждого результата ответа (а их как минимум два — верно/неверно) или диапазона результатов жестко указывается, какой вопрос будет следующим;

2) с отсутствием жестких связей. Здесь все вопросы делятся на группы, каждой из которых присваивается свой коэффициент сложности. При верном ответе на вопрос следующий будет задан из группы, коэффициент сложности которой выше, чем у текущей группы, при неверном ответе — из группы с более низким коэффициентом.

Могут существовать комбинированные варианты, когда связи есть, но они не привязаны жестко к одному вопросу, а ссылаются на группу вопросов, из которой один выбирается случайным образом.

Очевидно, что схема данных такой базы вопросов будет представлять собой достаточно сложное дерево или граф с множеством видов отношений узлов и групп узлов (рис. 2). Преимущества использования визуального проектирования здесь те же, что и перечисленные выше.

Аналогично система разработки тренажерных программ может использовать средства проектирования для разработки алгоритма тренажера. В общем случае алгоритм тренажера может содержать тот же набор связей и блоков, что и алгоритм обычной программы на структурном языке. О способах использования средств проектирования при реализации алгоритмов будет сказано ниже.

Кроме того, различные компоненты учебного курса могут взаимно вызывать друг друга, т.е. на самом деле все эти схемы данных в рамках учебного курса образуют одну сложную гиперсхему. Становится ясно, что средство редактирования не должно реализовываться в отдельных средствах для разработки различных видов КУП. Это должен быть отдельный модуль, реализованный в виде библиотеки. Гибкость настроек должна позволять использовать его для проектирования данных в любом виде КУП (посредством встраивания) и связывать эти данные между собой. С этим связаны сложности использования сторонних средств проектирования (не говоря уже о правовых аспектах вопроса, ведь эти средства будут применяться в коммерческих разработках учебных курсов, имеющих большие тиражи). Поэтому было принято решение разрабатывать собственный продукт.

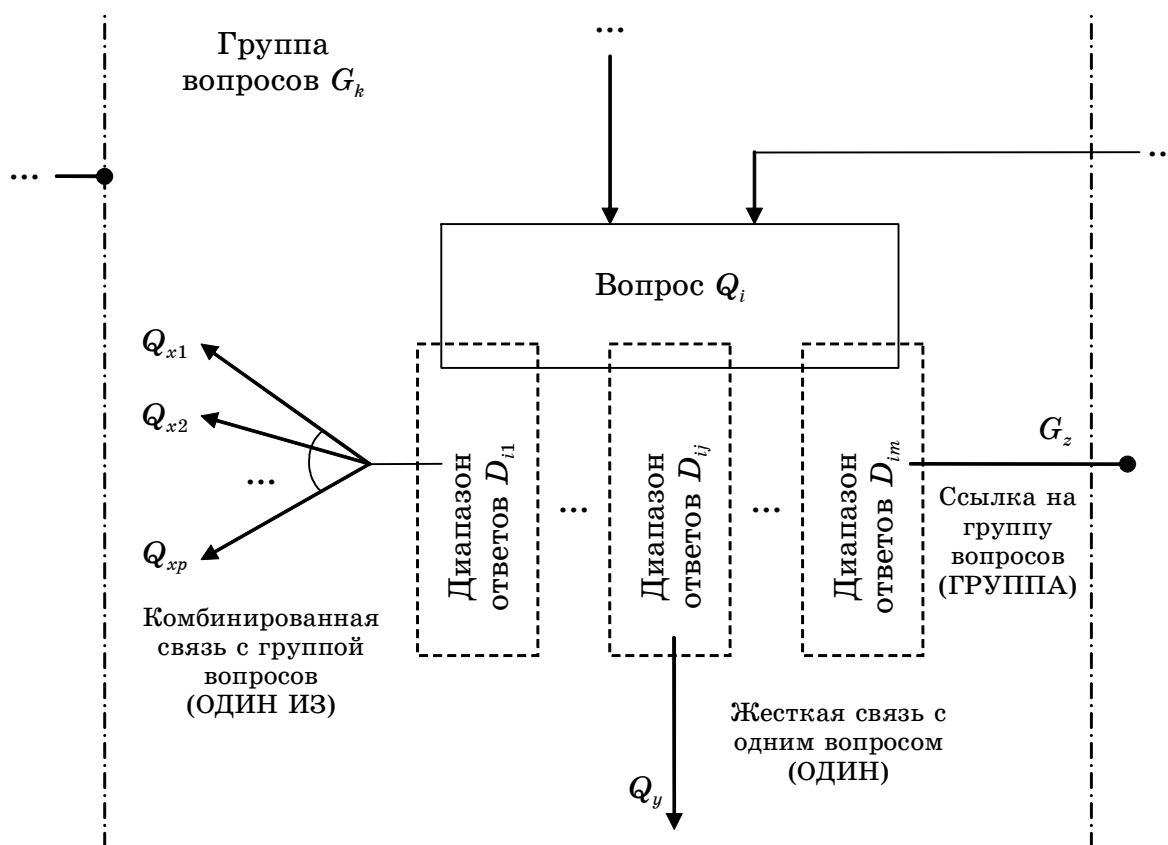


Рис. 2. Участок схемы данных САКЗ

Использование средств проектирования при разработке алгоритмов

Анализируя рис. 1 и 2, можно видеть, что приведенные схемы имеют много общего с графическим представлением алгоритма программы (блок-схем). Алгоритм также является направленным графом с определенным набором блоков, т.е. визуальную часть средств проектирования можно использовать для создания структуры алгоритмов и наполнения блоков соответствующими операциями.

Отметим, что различные редакторы блок-схем уже существуют. Например, это FlowChart Builder российской компании UNN [6]. Несмотря на ряд интересных идей (таких, например, как возможность использования пакета для документирования программ), нельзя не отметить недостатки (в контексте поставленной задачи):

- продукт используется в основном только для составления алгоритмов, хотя напрашивается решение создания более гибкой среды проектирования;
- синтаксис ориентирован только на один язык, напоминающий Pascal;
- отсутствуют средства трассировки алгоритмов;
- невозможно встроить продукт в рамки тестирующих программ и других интерактивных КУП.

Есть и другие продукты. Однако у всех из них существуют свои недостатки, препятствующие эффективному использованию в образовательном процессе. А именно, помимо упомянутых выше технических, имеются в виду следующие функции.

1. Проблема автоматизированного тестирования программ, написанных учащимися, состоит в том, что мы вынуждены рассматривать программу как черный ящик [5]. То есть мы можем проконтролировать, что выходные данные соответствуют входным, но каким образом они были получены? Был ли использован требуемый метод? В большинстве случаев ответить на этот вопрос невозможно. Если же вместо программы использовать блок-схему, то ее анализ значительно упрощается.

2. Блок-схемы с некоторыми условными операциями (т.е. не относящимися ни к какому конкретному языку программирования) можно использовать при решении задач, требующих проявления алгоритмических навыков. При этом знание конкретных языков программирования не обязательно. Такие учебные программы можно использовать на некоторых гуманитарных специальностях, а также в таких дисциплинах, как математика, физика и т.п. В них процесс решения задач, в общем случае, также представляет собой некий алгоритм.

3. Средства визуализации блок-схем можно использовать для построения изображения иллюстраций (даже интерактивных) в методическом материале и для просмотра алгоритма решения задачи в тренажерных программах, а средства их редактирования — для контроля знаний по широкому спектру дисциплин.

4. Несложно создать компонент, преобразующий построенную блок-схему в текст программы на выбранном языке программирования.

Рассмотрим еще одну проблему. Когда мы говорим о решении математических, физических и других задач людьми, не знакомыми с программированием, появляется проблема — как им вводить в блок-схему формулы? Ведь не каждый сможет написать что-то вроде

$$(\text{pow}(x,3) + \ln(\text{pow}(x,2) + 1))/(2 + \sin(3 \cdot x)),$$

хотя это соответствует всего лишь следующему выражению:

$$f(x) = \frac{x^3 + \ln(x^2 + 1)}{2 + \sin 3x}.$$

Рассмотрим решение этой проблемы.

Ввод математических выражений в естественном виде

Очевидно, что необходим компонент (также встраиваемый, чтобы его можно было использовать во всех видах КУП) для ввода математических выражений в естественном виде, позволяющий отображать также эти формулы на кадрах учебника и вычислять введенное выражение. Самые распространенные редакторы формул — Equation и MathType для пакетов MS Office, не обладают такой функциональностью. И нет гарантии, что они установлены на машине удаленного пользователя (например, при дистанционном обучении). Мощные пакеты типа MathCAD, MATLAB использовать для этих целей нерационально, к тому же их имеет еще меньшее число пользователей. Существуют и специальные продукты. В частности, при разработке компьютерного учебника «Высшая математика. Дифференциальное исчисление и интегральные уравнения» [7] для ввода формул использовалась программа Formulator компании Hermitech [8] (рис. 3).

Однако при использовании стороннего продукта были выявлены его недостатки (опять же относительно поставленных целей). В будущем будет разработан свой компонент, обладающий всей необходимой функциональностью.

В настоящее время на кафедре АСУ ТУСУРа внедряется групповое проектное обучение (ГПО). В рамках этой концепции каждый студент, начиная с третьего курса, должен входить в группы из 2–5 человек, совместно работающих над одним достаточно крупным

проектом. Автор руководит одной из таких групп (состоящей на данный момент из двух человек), перед которой и поставлена цель — реализовать подобную интерактивную систему для работы с математическими выражениями в естественном виде. После соединения двух данных технологий (т.е. компоненты для составления блок-схем, построенных на базе универсальной среды проектирования данных, и компоненты интерактивной работы с математическими выражениями) полученный продукт будет обладать уникальной совокупностью характеристик, выделяющих его на фоне распространенных аналогов.

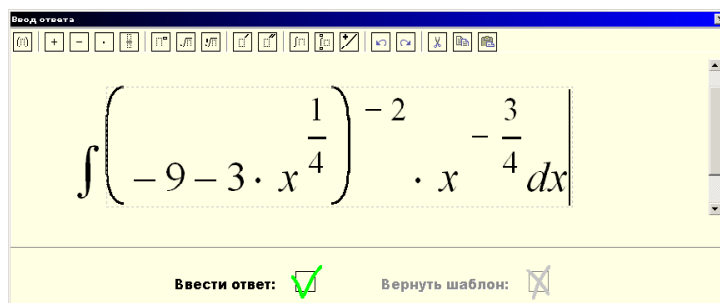


Рис. 3. Ввод формул в компьютерном учебнике «Высшая математика»

Будут набраны и другие группы для решения различных задач, связанных с автоматизацией разработки КУП.

Работа выполняется при поддержке гранта Президента РФ для государственной поддержки молодых российских ученых — кандидатов наук МК-9734.2006.9.

Литература

1. Автоматизация разработки компьютерных учебных программ / А.А. Мицель. — Томск : НТЛ, 2005. — 383 с.
2. Буч Г. UML. Руководство пользователя / Г. Буч, Д. Рамбо, А. Джекобсон. — М. : ДМК, 2000. — 432 с.
3. Проектирование компьютерных обучающих программ [Электронный ресурс] / Н.Н. Горлушкина [и др.]. — Режим доступа: <http://de.ifmo.ru/library/0055>.
4. Перегудов Ф.И. Основы системного анализа / Ф.И. Перегудов, Ф.П. Тарасенко. — Томск : НТЛ, 2001. — 396 с.
5. Веретенников М.В. Требования к системе автоматизированного контроля знаний / М.В. Веретенников // Тр. науч.-техн. школы-семинара студентов, аспирантов и молодых специалистов «Информационные системы». — Томск : Томск. гос. ун-т систем упр. и радиотехники, 2003. — С. 148-152.
6. Конструктор блок-схем FlowChart Builder [Электронный ресурс]. — Режим доступа: <http://www.unn.ru/vmk/graphmod>.
7. Клык В.В. Интерактивные компьютерные тренажеры по интегральному исчислению и дифференциальным уравнениям / В.В. Клык, А.А. Ельцов, К.Г. Шатлов // Известия ТПУ. — 2006. — №2. — С. 255-260.
8. Hermitech. Официальный сайт лаборатории [Электронный ресурс]. — Режим доступа: <http://www.hermitech.ic.zt.ua>.

Романенко Владимир Васильевич

Канд. техн. наук, доцент кафедры автоматизированных систем управления ТУСУРа
 Телефон: (3822) 41 34 54
 Эл. почта: voverkill@asu.tusur.ru

V.V. Romanenko

Using a data architecture management tools for development the computer educational course components

An article deal with the methods of using automated management tools of data structures visual designing for development of various interactive components in the computer educational programs (trainers, testing systems etc.).
