

УДК 37:004

К.Г. Шатлов

Редактор графов и блок-схем

Рассматривается решение проблемы отсутствия стандартных средств для визуального редактирования, отображения и хранения, данных, представляемых графами и блок-схемами.

Современный уровень развития технического и программного обучения предоставляет широкие возможности использования компьютеров в сфере образования. Электронное обучение является активной формой образования, так как позволяет обучающемуся непосредственно участвовать в познавательном процессе, и осуществляется с помощью различных тренажеров, виртуальных лабораторий и ряда подобных программ. Компьютерные тренажеры дают возможность достаточно быстро сформировать у обучающихся навыки по решению различных задач, освоить практический материал учебных курсов.

В настоящее время на кафедре автоматизированных систем управления создан ряд электронных курсов по различным техническим дисциплинам. Все эти дисциплины разработаны с помощью комплекса EduCad [1, 2], который постоянно развивается и дополняется новыми функциями и возможностями.

На текущий момент планируется разработка системы автоматического создания тренажеров по математическим дисциплинам, новой системы адаптивного траекторного обучения [3] и модификация EducadTextBook [2], в которой необходим графический редактор связей между уроками. Большинство математических задач, навыки решения которых будут вырабатываться тренажерами, могут быть описаны четкими алгоритмами. Самым простым и наглядным средством представления алгоритмов являются блок-схемы. Нарисовав блок-схему, разработчик может определить полностью логику работы всего тренажера, в идеале ему не придется писать ни одной строки кода.

Задачи математического и компьютерного моделирования очень часто связаны с решением задач, алгоритмы которых представляются в виде графов. Также имеется ряд различных прикладных задач, так или иначе связанных с графами и их частным случаем — блок-схемами. Если же программно хранить данные графы не представляет особой сложности, так как легко осуществляется с помощью динамических списков, то графически их визуализировать затруднительно. Стандартные средства ОС MS Windows предлагают только компоненты для визуализации линейных списков и иерархических деревьев в стиле программы «проводник». Рисовать блок-схемы позволяет лишь редактор Word, но использовать его средства в своих программах затруднительно, так как работа с ним возможна лишь посредством технологии OLE. Сторонние специализированные средства Microsoft Visio, Rational Rose, Flowchart builder [4] предлагают пользователю мощные графические средства для редактирования и просмотра, экспорт в растровые и векторные графические форматы, но в них не предусмотрены интерфейсы для встраивания в собственные разрабатываемые приложения. Неплохим вариантом является Delphi-компонент TBlocksHolder отечественной программы VisiPro [5], но он не позволяет рисовать соединительные стрелки произвольной формы, связывать с элементами блок-схем дополнительные данные (например, программный код, соответствующий действию, изображенному на блок-схеме) и формат сохранения данных VPS, хранит лишь данные о графическом представлении блок-схем и не дает возможности установить логическую взаимосвязь между блоками. Таким образом, требуется разработать компоненту для визуализации и редактирования графов.

Задачу можно разбить на три подзадачи:

- 1) разработать компоненты для визуализации узлов и соединительных линий;
- 2) разработать редактор графов;
- 3) разработать компонент визуализации графов.

Взаимосвязь разрабатываемых частей представлена на рис. 1.

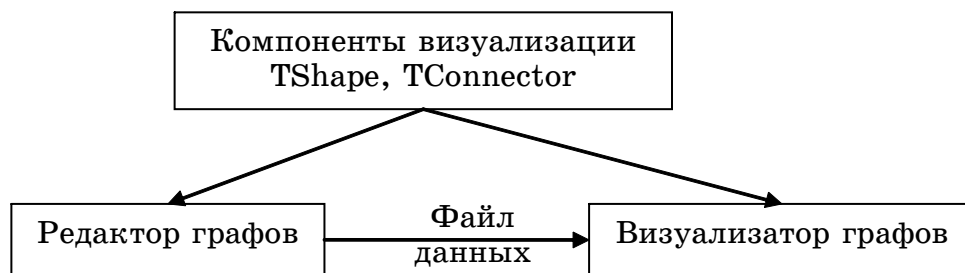


Рис. 1. Взаимосвязь разрабатываемых частей

Данная компонента планируется применяться для визуализации следующих данных.

Основное назначение:

- блок-схемы — блоки различной геометрической формы и стрелки между ними для описания алгоритмов (рис. 2,а).

Специфичное применение в рамках среды EduCad:

- редактор связей страниц — графические блоки с изображениями страниц;
- редактор связей вопросов — для определения связей в системах контроля знания с траекторийным движением.

Дополнительное применение:

- математические графы — графическое представление графов в виде кружков (узлов) и дуг (связей) между ними (рис. 2,б).
- представление моделей данных — базы данных, отображение атрибутов таблиц и их взаимосвязей (рис. 2,в).
- структурные схемы — представление различных структурных схем (электросхемы, модели данных).

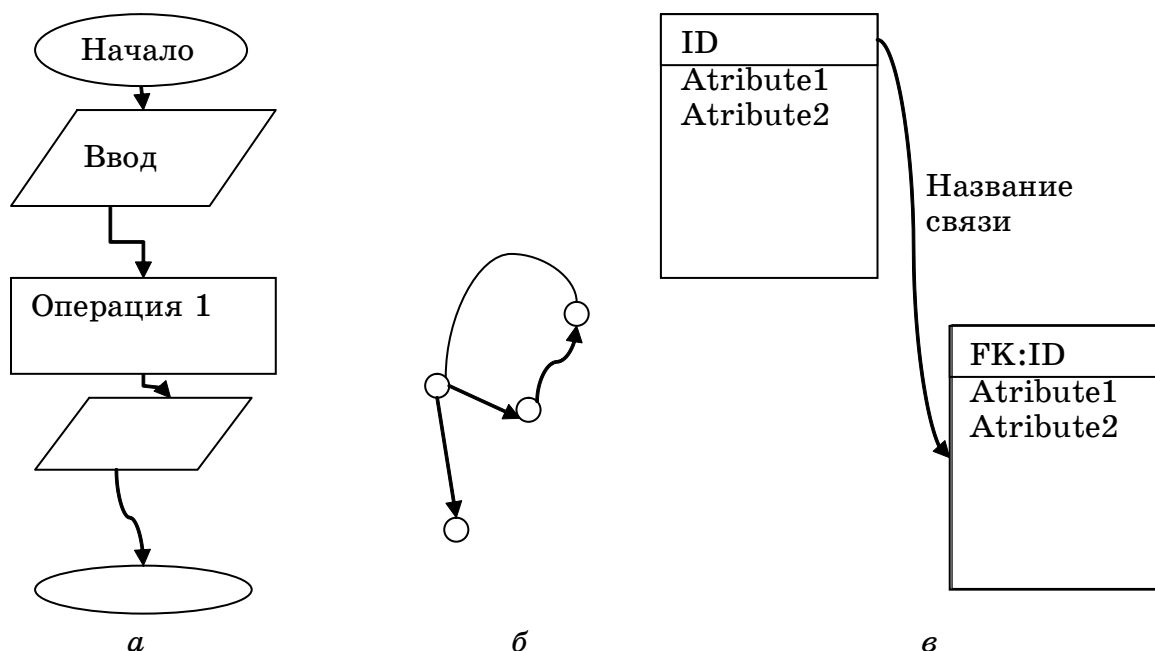


Рис. 2. Сферы возможного применения:

а — представление блок-схем; б — представление графов; в — представление моделей данных

Проанализировав имеющиеся аналоги и сферу применения, были сформированы следующие требования.

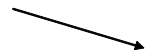


1. Поддержка разных типов узлов. Разрабатываемая компонента должна позволять отображать как стандартные геометрические фигуры (круг, овал, прямоугольник, наклонный

параллелепипед, ромб), так и позволять пользователю задавать полигоны произвольной формы путем перечисления относительных координат всех точек. Кроме того, содержать событие OnPaint для предоставления пользователю возможности полностью написать свой код визуализации блока.

2. Фиксированные точки входа/выхода. Стандартные блоки, используемые в редакторе, должны иметь фиксированное расположение точек входа/выхода соединительных линий, для того чтобы придерживаться определенного стандарта оформления согласно ГОСТу.

3. Подключаемая библиотека блоков. Редактор должен позволять подключать отдельно разработанные блоки визуализации узлов (на уровне Dynamic Link Library или файлов описаний формы блоков и точек входов/выходов).

4. Поддержка разных типов соединительных линий (стрелок). Разрабатываемая компонента должна позволять отображать стрелки трех базовых типов, а также содержать событие OnPaint для предоставления пользователю возможности полностью написать свой код визуализации стрелки.

Прямые линии (Straight)	
Линии, преломляющиеся под прямым углом (Elbow)	
Кривые (Curved)	

5. Неограниченность числа соединений блока. В каждый блок может как входить, так и выходить несколько стрелок.

6. Группировка. Должна быть возможность группировки нескольких блоков в один связанный блок для облегчения редактирования.

7. MDI интерфейс редактора. Разрабатываемая система должна позволять редактировать одновременно несколько документов в одном окне.

8. Многоуровневость. Довольно часто бывает, что блок-схемы имеют большой размер и не помещаются на экран, поэтому необходима возможность объединения, сворачивания нескольких блоков в один агрегатный блок-гиперблок. На каждом уровне должно быть представлено не более 15 структурных блоков. Соответственно средства просмотра должны обеспечить декомпозицию крупных блоков на более мелкие для рассмотрения детальной структуры и взаимосвязей. При клике мышью на гиперблок должно открываться новое окно с его декомпозицией.

9. Интерфейс взаимодействия. Разрабатываемые компоненты должны иметь набор стандартных событий на действия пользователя, таких как OnClick/OnDblClick (клик/двойной клик мышью); базовых свойств, например Caption (текст внутри блока), Color (цвет блока).

10. Поддержка UML. Возможность экспорта и импорта в формат UML.

11. Агрегация данных. Компоненты графического представления могут быть сразу объединены с данными, например, компонент параллелепипед, используемый в блок-схемах, может содержать сразу программный код, соответствующий алгоритму, а, например, в системе контроля знаний в нем может содержаться сразу текст вопроса.

В качестве среды разработки был выбран Borland C Builder 4 для обеспечения совместимости с ранее разработанными компонентами комплекса EduCad.

На данный момент компонент визуализации реализован в следующем виде: от стандартного класса *TGraphicControl* унаследованы два класса — *BaseBlock* (блок) и *BaseArrow* (стрелка).

Свойства класса *BaseBlock* и возможные значения:

- ID — уникальный идентификатор блока;
- Type=btSquare, btRomb, btCustom, btAggregate — тип блока (квадрат, ромб, пользовательский, гиперблок);

- Color, FrameColor — цвет блока и рамки блока;
- Caption — текст внутри блока;
- OnClickEvent — событие на нажатие мыши;
- Pins[] — массив, описывающий входы/выходы из блока.

Свойства класса типа стрел и возможные значения:

- ID — уникальный идентификатор стрелки;

- Type=taSimple,taDuplex,taCustom — тип направления стрелки (однонаправленная, двунаправленная, пользовательская);
- GraphicType = tgStraight, tgElbow, tgCurved, tgCustom — тип формы стрелки (прямая, преломляющаяся под прямым углом, кривая Безье);
- Caption — текст подписи к стрелки;
- InPin, OutPin — указатели на блоки, которые соединяет стрелка.

Экземпляры данных компонентов размещаются на компоненте *TScrolBox*, а их редактирование осуществляется с помощью специально написанного компонента *TControlHandler*, позволяющего выбирать элементы и менять их размеры. Разработан простой одноуровневый редактор, который дает возможность добавлять, удалять, перемещать, изменять размеры, цвета и надписи к прямоугольным блокам, а также связывать их однонаправленными связями. Сохраняется только визуальная часть в формате dfm(Delphi Form). Информационную часть о связях между блоками планируется хранить с некоторой избыточностью, сохраняя данные как о связях самих блоков (у каждого блока будут описаны ID блоков, с которыми он связан), так и о связях стрелок (у каждой стрелки будут описаны ID блоков, с которыми она связана), в дальнейшем такое хранение упростит представление графов в памяти компьютера в виде динамических деревьев.

В ближайшее время будет осуществлена реализация всех заявленных в статье функциональных возможностей программы. В дальнейшем планируется создание на этой основе двух специализированных систем: автоматического проектирования и создания математических тренажеров с мощными инструментальными средствами визуальной разработки диаграмм и системы контроля и проверки знаний.

Литература

1. Романенко В.В. Общие принципы автоматизации разработки компьютерных учебных пособий в комплексе EduCAD / В.В. Романенко // Программные продукты и системы. — 2003. — № 2. — С. 34–37.
2. Автоматизация разработки компьютерных учебных программ / А.А. Мицель. — Томск : НТЛ, 2005. — 384 с.
3. Романенко А.В. Контроль знаний в электронных обучающих комплексах / А.В. Романенко // Современное образование: традиции и новации: материалы всероссийской научно-методической конференции, Россия, Томск, 2-3 февраля 2006 г. — Томск : Томск. гос. ун-т систем упр. и радиоэлектроники, 2006. — С. 174-175.
4. Сайт программы «Конструктор блок-схем», разработанной на кафедре интеллектуальных информационных систем и геоинформатики Нижегородского государственного университета им. Н.И. Лобачевского НИИ прикладной математики и кибернетики [Электронный ресурс]. — Режим доступа: <http://www.cir.nnov.ru/vmk/graphmod/index.php?id=fchb>
5. Официальный сайт программы создания блок-схем «VisiPro» [Электронный ресурс]. — Режим доступа: <http://visipro.nm.ru/>.

Шатлов Константин Геннадьевич

Аспирант кафедры автоматизированных систем управления ТУСУРа

Телефон: +7 913 818 48 09

Эл. почта: frodo@ms.tusur.ru

K.G. Shatlov

Editor of graph and flowcharts

Absence problem decision of the standard tools for visual editing, visualization and storing flowcharts is reviewed in the article.
