

Основные принципы разработки визуального конструктора программ имитационных моделей

А. С. Помаз

Введение

В связи с приходом на рынок программного обеспечения операционных систем с графической оболочкой и постепенным вытеснением аналогичных продуктов с интерпретатором командной строки перед разработчиками моделирующего инструментария появились следующие цели [1]:

- создание новых средств имитационного моделирования с использованием всего сервиса разработки программного обеспечения, предоставляемого новыми операционными системами;
- поиск новых подходов к моделированию объектов, имеющих сложную структуру и взаимодействие между структурными составляющими.

Данные две цели, можно сказать, являются парадигмами современного моделирующего инструментария.

Одним из программных средств из класса моделирующего инструментария, отвечающих указанным парадигмам, является визуальный конструктор программ имитационных моделей, который разрабатывается в комплексе с системой моделирования MICIC [2]. Рабочее название визуального конструктора – M-BUILDER.

M-BUILDER предполагается как средство автоматизации этапа создания программы модели для системы моделирования MICIC. То есть можно говорить о технологическом комплексе <M-BUILDER + MICIC>.

Основные положения концепции приложения M-BUILDER заключаются в следующем:

- Использование визуальных приемов в большинстве технологических операций программного средства.
- Двойственное представление структуры имитационной модели.
- Генерация программы имитационной модели на основе спроектированной визуальной структуры модели.

1. Визуальность как основополагающий принцип конструктора имитационных моделей

Любая умственная деятельность человека связана с оперированием зрительными сущностями. Однако это требует значительных усилий от специалиста, выполняющего такого рода работу. В сфере информационных технологий для более быстрого получения конечного результата стали использоваться приемы и технологии, которые позволяют оперировать зрительными образами, некоторым способом соотносящиеся с физическими сущностями. В качестве примеров можно привести технологию WYSIWYG, методики работы в CAD-системах, ставшей уже классической технологию “drag-n-drop”, компьютерные системы анимации типа Maya [3].

Визуальность является основой множества технологических операций в программном средстве M-BUILDER. Данное положение концепции имеет место при выполнении следующих технологических этапов:

- разработка структуры концептуальной модели;

- проектирование механизмов обслуживания.

Концептуальная модель – это результат выполненной декомпозиции сложной системы и сопутствующих ей других операций при реализации имитационной модели [4]. Будем полагать, что структурная составляющая концептуальной модели – это также модель, а взаимодействие между структурными составляющими – связь между моделями. Далее, соотнесем модели визуальный образ, который в техническом смысле есть геометрическая фигура либо растровое изображение. Геометрическую фигуру наделим различными атрибутами, позволяющими идентифицировать любой элемент модели по его начертанию. Аналогично связи между моделями соотнесем ломаную линию с атрибутами отображения. Визуально ломаная линия соединяет два графических образа, отождествленные с моделями. Связи между моделями могут быть направленными, что отображается направляющей стрелкой.

Рассмотрим вопрос отображения структуры концептуальной модели. Для размещения визуальных образов и ломаных линий, их соединяющих, используется такой визуальный объект как окно, являющееся фундаментальной единицей отображения информации в операционной системе. Внесение и удаление визуальных элементов в окно, как в контейнер, соединение образов моделей ломаными линиями как связями и составляет сущность проектирования структуры концептуальной модели. Программная подсистема, отвечающая за проектирование структуры, позволяет корректировать введенные ранее атрибуты визуальных элементов, то есть специалист имеет возможность управлять отображением. Пример разработки структуры концептуальной модели иерархической сложной системы (тестовая модель сети городского пассажирского транспорта [5]) приведен на рис. 1.

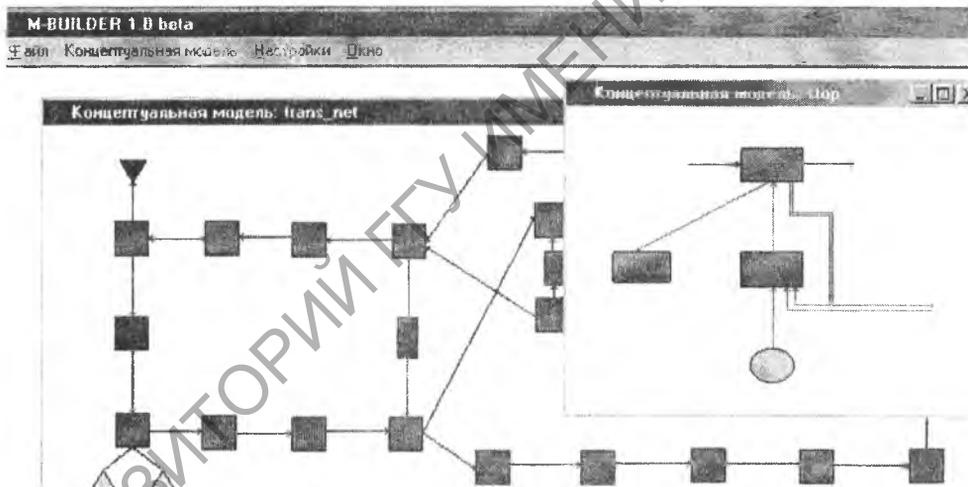


Рис. 1. Структура имитационной модели транспортной сети в M-BUILDER

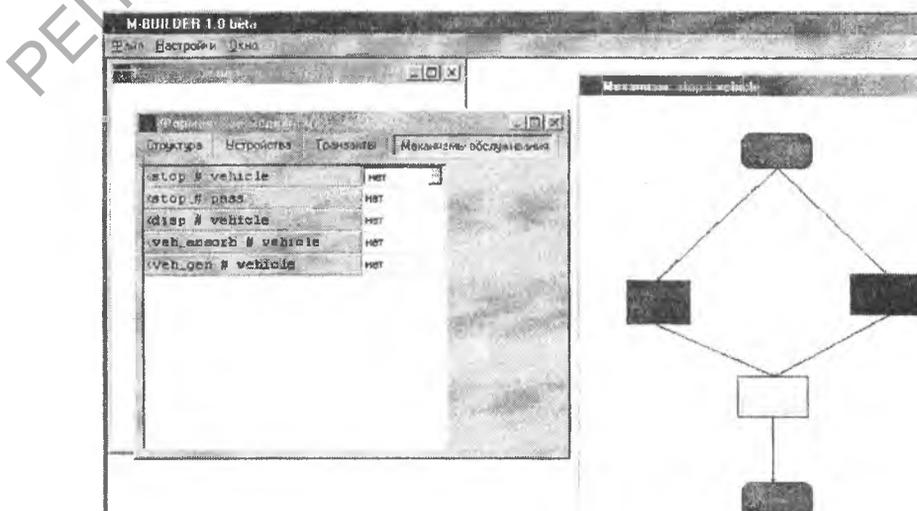


Рис. 2. Проектирование механизма обслуживания в M-BUILDER

Рассмотрим визуальное проектирование механизмов обслуживания в программном средстве. Будем полагать, что механизм обслуживания – это ориентированный граф [6], где вершины – это активности, а дуги – передача управления между активностями. Тогда мы можем соотнести активности некую геометрическую фигуру или графическое изображение, передаче управления – ломаную линию (рис. 2). Как и в случае структуры концептуальной модели, мы наделим визуальные элементы механизма обслуживания идентичными атрибутами.

Для визуального отображения механизма обслуживания также используется окно, в котором специалист группирует соответствующие визуальные элементы. В этом заключается сущность визуализации механизма обслуживания.

2. Двойственность в представлении структуры модели

Будем полагать, что модель на концептуальном уровне является узлом. То есть модель имеет в общем случае внутреннюю структуру, состоящую из других моделей и соединяющих их связей. Модели, входящие в узел, могут быть узлами в свою очередь. Таким образом, мы получаем дерево узлов, где верхний уровень – это сама концептуальная модель, нижний уровень – модели, не имеющие собственной внутренней структуры. В дальнейшем такие модели мы будем называть элементарными. Отсюда вытекает тот факт, что структура модели представима иерархически в виде дерева.

Выше уже отмечалось, что модель имеет внутреннюю структуру, состоящую из моделей и соединяющих их связей (рис 1). Очевидно, что такая структура данных с точки зрения теории графов является сетью. Таким образом, мы получаем сетевое представление структуры модели на концептуальном уровне.

Следовательно, двойственность в представлении структуры одной и той же имитационной модели на концептуальном уровне является вторым ключевым аспектом при разработке визуального конструктора M-BUILDER. Технологически это означает наличие более развитого контроля над структурой модели и предупреждение ошибок в структуре, что позволяет минимизировать ошибки в генерируемом модуле программы имитационной модели.

3. Генерация программы имитационной модели

Приведем последовательность этапов перехода от спроектированной визуальной структуры модели к программе имитационной модели. Первый технологический этап – проектирование структуры модели. Специалист составляет концептуальное описание объекта моделирования, выделяет отдельные модели как структурные составляющие, соединяя их между собой связями. Связь в общем случае также является узлом и может быть направленной. В контексте визуального конструктора мы получаем спроектированную визуальную структуру модели.

Далее осуществляется переход от концептуальной модели к формальной. Такие сущности как модели мы заменяем устройствами, определяя их режим функционирования. Связи в данном случае соответствуют путям для перемещения транзактов.

Переход от концептуальной модели к формальной осуществляется по следующим правилам:

- Из всех моделей в структурном смысле выделяется множество устройств модели.
- Посредством анализа дерева структуры все связи между моделями конвертируются в множество транзактов. В данном случае играют роль визуальные атрибуты связей, то есть в первую очередь унифицируются совокупности визуальных атрибутов, которые затем соотносятся с транзактами.
- Строится дерево версий устройств путем прохода всей иерархии структуры, причем в него включаются и структуры связей.
- При наличии импортированной концептуальной модели в формальную определяют

ся системные параметры устройств и транзактов.

Формальная модель была бы неполной, если бы не создавалась средствами визуального конструктора совокупность механизмов обслуживания. Для этого в программе введена подсистема, обеспечивающая визуальное проектирование механизмов обслуживания. В данном случае визуальные образы соотносятся с активностями. Для определенности полагается наличие начальной и конечной активностей. Связи означают квазипараллельную передачу управления. Для визуализации механизмов обслуживания были взяты принципы, положенные в ЕСПД (ГОСТ 19.003-80). Теоретические аспекты механизмов обслуживания и обоснование возможности их визуализации рассмотрены в [7].

Когда определены все компоненты формальной модели, дерево версий устройств и механизмы обслуживания, происходит собственно процесс генерации программы модели. Прежде всего, необходимо сказать, что сам визуальный конструктор не содержит описание выходного языка, которым описывается программа модели. Для этой цели было введено понятие языкового модуля, в котором описан язык генерации программы. По существу, это динамически подключаемая к визуальному конструктору библиотека, которая принимает в качестве входных параметров формальную модель в определенном формате и получает управление. На выходе языкового модуля – программа имитационной модели.

Так как языковой модуль подключается динамически к визуальному конструктору, то отсюда вытекает преимущество их использования – возможность генерации программ под различные системы моделирования без внесения изменений в сам конструктор. Такой подход обеспечивает большую функциональность программного средства.

Заключение

Мы рассмотрели три фундаментальных принципа создания нового представителя моделирующего инструментария. Можно сказать, что визуальность используется практически на всех этапах создания программы модели. Визуальный конструктор обеспечивает достаточно развитую работу со структурой модели, что повышает уровень работы с концептуальной моделью.

Была приведена схема создания программы модели в целом и рассмотрены теоретические аспекты разрабатываемой технологии. Концепция языкового модуля – одно из новшеств программного средства, повышающее значимость данного моделирующего инструмента. Гибкость подсистемы генерации делает визуальный конструктор инвариантным от системы моделирования, возлагая решение проблемы генерации на разработчиков языковых модулей.

Abstract

Tree basic conceptions of the application of M-BUILDER intended for the visual creation of a simulation model program are considered in this article. It ensures a sufficiently advanced work with a model structure. The scheme of a model program performance is given. Theoretical problems of the provided technology are described.

Литература

1. Гульяев А.К. MATLAB 5.2. Имитационное моделирование в среде Windows: Практическое пособие. – СПб.: КОРОНА принт, 1999. – 288 с.
2. Задачи и модели ИСО. Ч. 3. Технология имитации на ЭВМ и принятие решений: Уч. пособие / И.В. Максимей, В.Д. Левчук, С.П. Жогаль, В.Н. Подобедов. – Гомель: БелГУТ, 1999. – 150 с.
3. Основы информатики: Учеб пособие / Под ред. А.Н. Морозевича. – Мн.: Новое знание, 2001. – 544 с.
4. Максимей И.В. Имитационное моделирование на ЭВМ. – М.: Радио и связь, 1988. – 232 с.

5. Левчук Е.А. Технологические этапы имитационного моделирования системы общественного городского транспорта // Сб. трудов VI Междунар. науч. конф. «Актуальные проблемы информатики», Ч. 1. Мн.: БГУ, 1998. – С. 225-231.
6. Басакер Р., Саати Т. Конечные графы и сети. – М.: Мир, 1973. 368 с.
7. Помаз А.С., Левчук В.Д. Пользовательские уровни рассмотрения объекта моделирования в визуальном конструкторе M-BUILDER // Тезисы докладов V Республиканской научно-технической конференции студентов и аспирантов “Новые математические методы и компьютерные технологии в проектировании, производстве и научных исследованиях”. – Гомель, 2003. – С. 76-77.

Гомельский государственный
университет им. Ф.Скорины

Поступило 20.03.04

РЕПОЗИТОРИЙ ГГУ ИМЕНИ Ф. СКОРИНЫ