Электронный научно-технический журнал Октябрь 2011
http://www.bru.mogilev.by

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ВИЗУАЛЬНОГО ПОСТРОЕНИЯ ИМИТАЦИОННОЙ МОДЕЛИ[1]

В.Г. Маслаков, Д.М. Албкеират, **А.И. Якимов**

Аннотация. В статье представлена программная реализация визуального построения имитационной модели на основе компонентов. Приведены диаграммы вариантов использования и компонентов программного комплекса. Показаны основные формы программы и технология визуального составления имитационной модели.

Ключевые слова: имитационная модель, компонент, визуализация, программный комплекс.

1. Введение

Программа Designer является составной частью программно-технологического комплекса имитации сложных систем BelSim2#. Имитационная модель в программе Designer строится с помощью компонентов, которые представляют собой процессы или объекты. Каждый из компонентов может иметь произвольное количество входов и выходов, через которые могут передаваться данные, имеющие различное логическое значение.

Предлагаемый программный комплекс планируется использовать в имитационном моделировании [1] для визуального построения моделей с соединением точек передачи сообщений от одного компонента к другому, определяя логическое значение точек соединения в программном коде, задаваемом в описании компонентов. Предполагается, что комплекс будет использоваться для осуществления дискретно-событийного моделирования и исследования системной динамики.

2. Основные требования к функциям программы системы имитации

Требования, предъявляемые к программной реализации системы имитации:

1 Создание классов – требование к выходным данным, получаемым при построении модели. В итоге всей работы пользователя последний должен получить набор классов либо подключаемую библиотеку с классами компонентов;

2 Создание модели – подразумевается создание графического представления имитационной модели, состоящего из набора компонентов и связей между ними;

3 Создание компонентов – предоставление возможности создавать компоненты, являющиеся структурной единицей имитационной модели;

4 Установка связей – должна быть обеспечена возможность устанавливать связи между компонентами, соответствующие потокам перемещения данных между компонентами;

5 Создание свойств – каждый компонент должен иметь набор свойств в соответствии с характеристиками соответствующего объекта реального мира, относительно которых ведётся имитационное моделирование.

[1] Работа выполнена при финансовой поддержке ГПНИ по заданию «Информатика и космос 1.3.02»

6 Создание методов – каждый компонент должен иметь внутреннюю логику, реализуемую проектировщиком модели на языке C# [2].

3. Диаграмма вариантов использования

Вариант использования представляет собой последовательность действий, выполняемых системой в ответ на событие, инициируемое некоторым внешним объектом (действующим лицом). Разрабатываемая система предполагает наличие лишь одного участника – пользователя программы (рисунок 1).

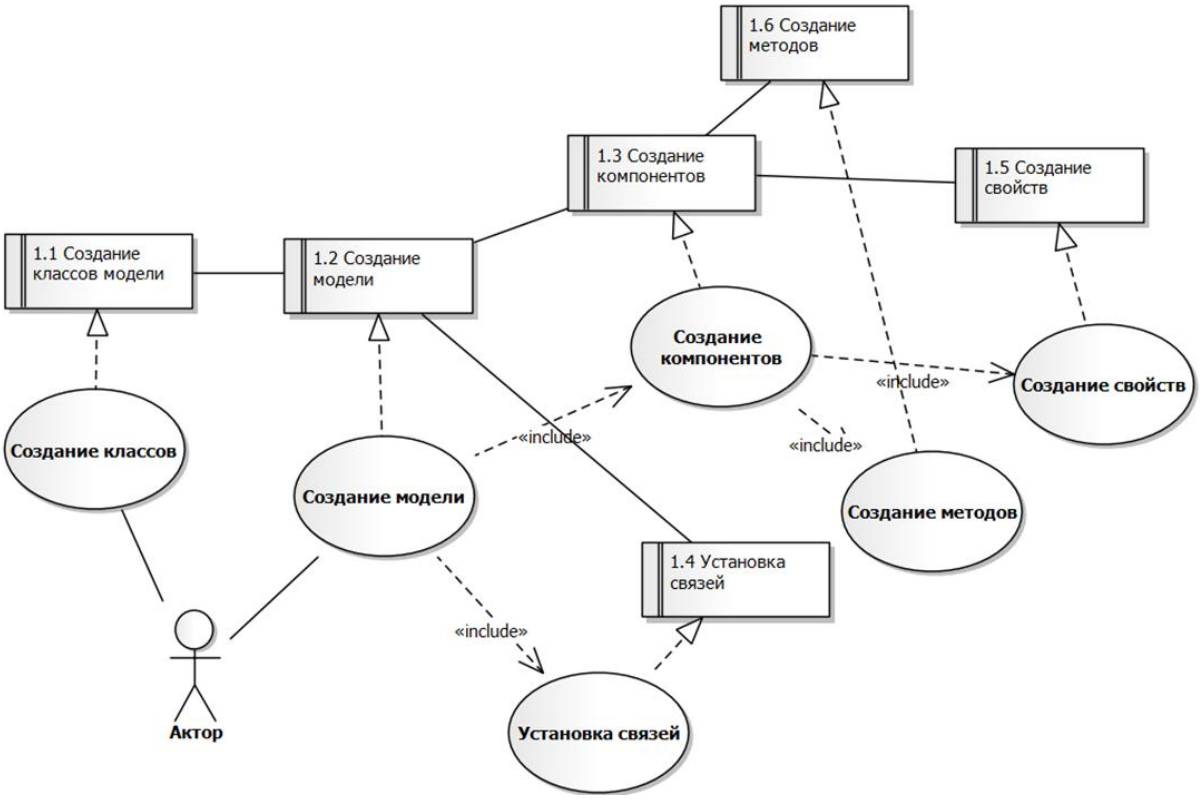

Рис. 1. Диаграмма вариантов использования

Деятельность пользователя включает создание модели, включающее создание компонентов, установку связей между компонентами, создание свойств и методов; создание классов.

4. Основные формы программы

Для организации ввода/вывода данных используются windows-формы. Всего разработано 5 форм: главное окно, которое включает в себя также дизайнер модели (отдельное MDI-окно), формы настроек компонента и агрегатора и окно сохранения шаблона компонента.

Главная форма содержит форму дизайнера и панель инструментов, с помощью которых можно управлять компонентами (т. е. добавлять и удалять компоненты, а также перемещаться по уровням).

Дизайнер модели лишь отображает структуру имитационной модели и содержит панели компонентов и их связи. Каждая панель компонентов имеет контекстное меню,

из которого можно выполнить действия, связанные с тем или иным компонентом (рисунок 2).

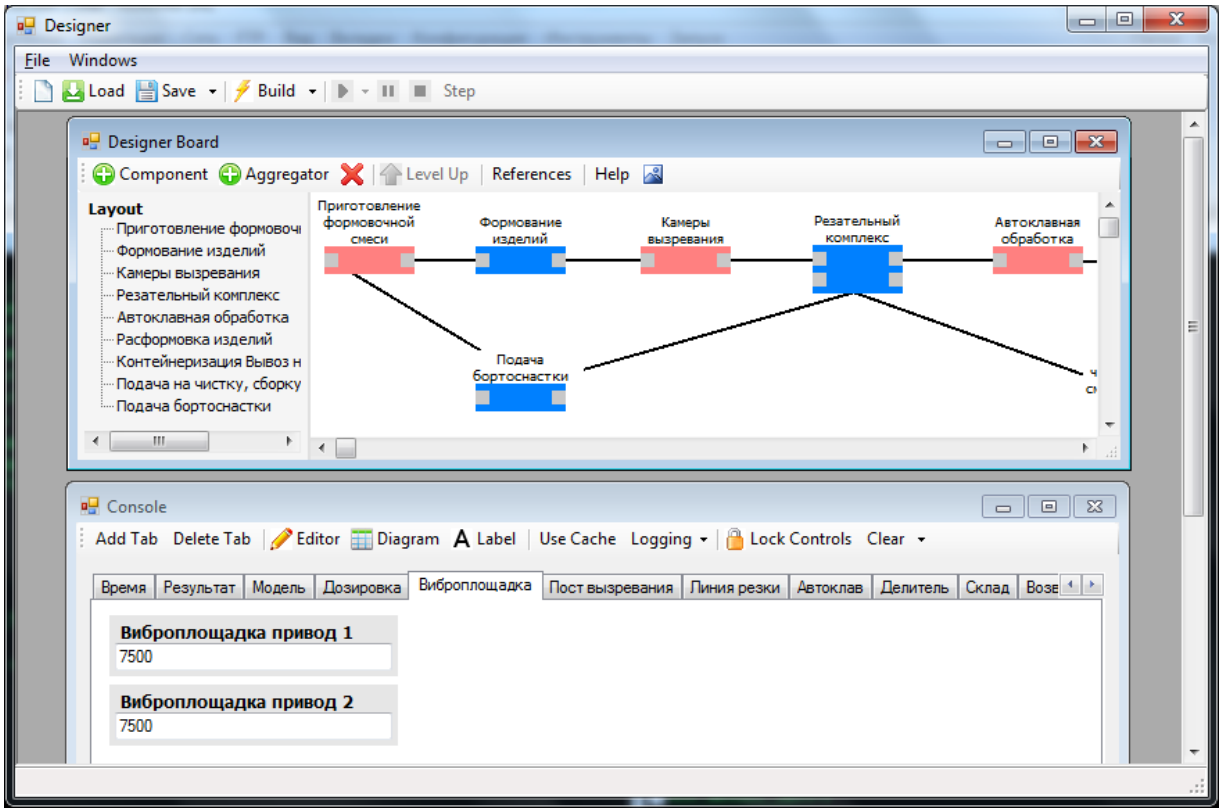

Рис. 2. Основные формы

Форма настройки компонента представляет собой интерфейс с вкладками для настройки входных и выходных точек соединения, настройки параметров компонентов и набора исходного кода компонента. Для настройки точек соединения и параметров используется контрол типа «таблица», который позволяет добавлять и удалять строки без использования дополнительных контролов (т. е. кнопок). Поскольку каждой строке соответствует название переменной, реализован контроль уникальности переменной.

Форма сохранения шаблона содержит лишь поле для ввода названия шаблона. Имя шаблона не должно быть уникальным, потому проверка введённого значения не производится.

Форма настройки агрегатора содержит список входящих и выходящих соединений низлежащих компонентов и предоставляет возможность выбрать то или иное соединения для вывода на верхний уровень дерева вложенности. Выбор точек пользователем может быть совершенно произвольным [3].

5 Составление имитационной модели

Описание модели начинается с формализации всех объектов и описания их в программном коде на языке C#. Поскольку программа имеет возможность сохранять шаблоны объектов, можно один раз запрограммировать шаблон и затем использовать его с различными параметрами. Составление модели происходит в окне Designer Board. С

помощью кнопки *Add Activity* на панели осуществляется добавление элементов в модель. После нажатия на кнопку появляется окно *Activity Setup*. Окно содержит следующие вкладки:

- I/O. В двух таблицах указываются входы и выходы компонента. Поле Name отвечает за описание поля, а Field Name – за указание имени поля. Это имя будет впоследствии использоваться в программном коде, так что оно должно иметь имя, соответствующее критериям описания идентификаторов, то есть, содержать только буквы, цифры и символ подчеркивания
- Parameters. Здесь в таблице указываются параметры этого компонента, которые являются общими для компонентов данного типа, но значения которых могут варьироваться от компонента к компоненту. В поле Name текстом описывается параметр, в Field Name указывается идентификатор, который будет использоваться в программном коде, а в Value указывается значение параметра. Если параметр должен будет использоваться как строка, его значение обязательно должно быть взято в кавычки, как и любая строка в C#.
- Code. В этом поле описывается программный код компонента.

На окне *Activity Setup* также содержится поле Name, в котором указывается описание объекта, и поле *Component Class Color*, в котором указывается цвет элемента управления на окне Designer Board, соответствующего данному объекту. Поле Name обязательно должно быть уникально для всей модели. Цвета сохраняются в шаблоне, так что с помощью цвета можно разделять по цвету объекты разных типов (например, процессы и хранилища). Кнопка *Save Template* отвечает за сохранение шаблона, а Assign Template – за применение шаблона к данному модулю. Почти всегда при сохранении шаблона нужно указывать параметры, не определяя их значение, поскольку значения параметров будут индивидуальными для каждого объекта [4].

6 Диаграмма компонентов программного комплекса

Диаграмма компонентов – статическая структурная диаграмма, показывает разбиение программной системы на структурные компоненты и связи (зависимости) между компонентами. В качестве физических компонент могут выступать файлы, библиотеки, модули, исполняемые файлы, пакеты и т. п.

Расширенные возможности использования графических средств ввода и вывода данных эксперимента предоставляются платформой Microsoft .NET Framework. Для реализации пользовательского интерфейса создатель модели способен создать динамически подгружаемую библиотеку (рисунок 3), которая будет содержать необходимые поля ввода и вывода данных, причём их тип и представление не будет ограничиваться средой.

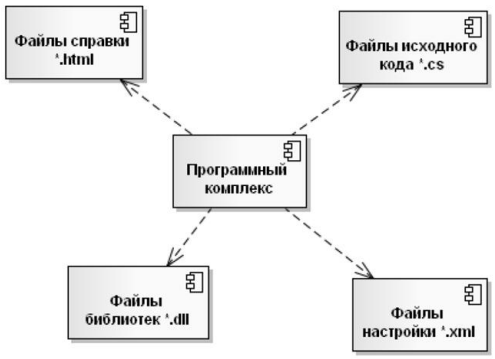

Рис. 3. Диаграмма компонентов программного комплекса

Для того, чтобы полностью абстрагироваться от среды моделирования, модель может быть скомпилирована в отдельную динамически подключаемую библиотеку без необходимости изменять саму модель. Таким образом, программист способен подключить библиотеку модели и моделирующее ядро системы моделирования к своему программному продукту без необходимости в дальнейшем ссылаться на дизайнер модели [5].

7. Заключение

Компонентный подход оправдал себя с разных сторон. Во-первых, визуальная модель значительно сокращает время на разработку каркаса модели, представляя модель или какой-то её уровень целиком на экране. Во-вторых, при имеющейся IDEF0-диаграмме не нужно переводить имеющуюся структуру диаграммы в программный код. В-третьих, при компонентном подходе компоненты оказываются изолированными один от другого, что позволяет вести разработку сразу командой. Используемая в системе платформа Microsoft .NET обеспечивает баланс между гибкостью, расширяемостью модели и быстродействием.

Практическое применение программный комплекс получил при визуальном построении имитационных моделей деятельности предприятий общественного питания [4], обувного производства [6].

Литература

1. **Якимов, А. И.** Технология имитационного моделирования систем управления промышленных предприятий : монография / А. И. Якимов. – Могилев: Белорус.-Рос. ун-т, 2010. – 304 с.: ил.
2. **Troelsen, A.** Pro C# 2008 and the .NET 3.5 Platform Fourth Edition / A. Troelsen. – New York: Apress, 2007. – 726 p.
3. **Маслаков, В. Г.** Визуальное построение имитационной модели на основе компонентов / В. Г. Маслаков, А. И. Якимов // Информационные технологии и системы 2011 (ИТС 2011): материалы междунар. науч. конф., БГУИР, Минск, Беларусь, 26 октября 2011 г. / редкол. : Л. Ю. Шилин [и др.]. – Минск : БГУИР, 2011. – С. 236–237.
4. **Маслаков, В. Г.** Визуальное программирование имитационных моделей производственных систем / В. Г. Маслаков, Д. М. Албкеират; науч. рук.: А. И. Якимов // Новые материалы, оборудование и технологии в промышленности : материалы междунар. науч.-техн. конф. молод. ученых; редкол.: И. С. Сазонов (гл. ред.) [и др.], Могилев, 18–19 ноября 2010 г. – Могилев: Белорус.-Рос. ун-т, 2010. – С. 163.
5. **Маслаков, В. Г.** Технология создания визуальных средств имитационного моделирования / В. Г. Маслаков; науч. рук.: А. И. Якимов, К. В. Захарченков // 46-я студенческая научно-техническая конференция Белорусско-Российского университета: материалы конф., редкол.: И. С. Сазонов (гл. ред.) [и др.]; 27 мая 2010 г. – Могилев: Белорус.-Рос. ун-т, 2010. – С. 142.
6. **Маслаков, В. Г.** Визуальные средства имитационного моделирования производственного процесса в ОАО «Обувь» / В. Г. Маслаков; науч. рук.: А. И. Якимов, К. В. Захарченков // 47-я студенческая научно-техническая конференция Белорусско-Российского университета: материалы конф., редкол.: И. С. Сазонов (гл. ред.) [и др.]; 26 мая 2011 г. – Могилев: Белорус.-Рос. ун-т, 2011. – С. 127.

Маслаков Владислав Геннадьевич
Выпускник кафедры Автоматизированные системы управления
Белорусско-Российский университет, г. Могилев
Тел.: +375(222) 25-24-47
E-mail: dreamer.mas@gmail.com

Албкеират Джихад Мохаммад
Аспирант кафедры Автоматизированные системы управления
Белорусско-Российский университет, г. Могилев
Тел.: +375(033)692-05-46
E-mail: albkerat@hotmail.com

Якимов Анатолий Иванович
Доцент кафедры Автоматизированные системы управления
Белорусско-Российский университет, г. Могилев
Тел.: +375(222) 25-24-47
E-mail: ykm@tut.by