

Визуальное построение имитационной модели на основе компонентов

Маслаков В.Г.; Якимов А.И.
Кафедра АСУ, электротехнический факультет
ГУВПО «Белорусско-Российский университет»
г. Могилев, Беларусь
e-mail: ykm@tut.by

Аннотация — Рассматривается метод визуального проектирования имитационной модели на основе компонентов. Представлена программная реализация визуального моделирования, технология проектирования визуальной модели.

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

I. ВВЕДЕНИЕ

В настоящее время существуют комплексы имитационного моделирования (MATLAB, AnyLogic, Arena, Business Studio, Powersim и др.) с предопределённой, не полностью контролируемой программным кодом логикой, либо с недостаточным разделением характеристик компонентов.

Предлагаемый программный комплекс планируется использовать в имитационном моделировании 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.

II. ОСНОВНЫЕ ПОЛОЖЕНИЯ ВИЗУАЛЬНОГО МОДЕЛИРОВАНИЯ

Одной из конечных целей работы программного комплекса визуального моделирования является создание документа, в котором будет отражен набор компонентов и его соединения (рис. 1).

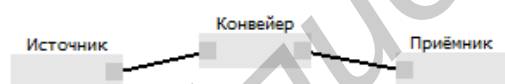


Рис. 1. Визуальное представление модели

Документ отражает компоненты и потоки данных в модели. На рис. 1 приведены компоненты «источник», «конвейер» и «приёмник», которые являются объектами реального мира. Каждый из этих объектов имеет своё устройство и назначение, но каждый работает с сущностями, с которыми работает другой объект, что позволяет сформировать из компонентов законченную систему. В данном случае на рис. 1 представлена система передачи некоторой сущности из источника в приёмник посредством конвейера. Принимая во внимание, что целью моделирования является анализ состояния объекта в контексте целой системы, компонент системы имеет набор входов и выходов, через которые объект получает воздействия извне и сам оказывает воздействие на другие компоненты. Каждый вход и выход имеет описание и уникальный идентификатор, который будет фигурировать в программном коде модели, который реализует логику компонента. Используя графический

интерфейс программы, входы и выходы компонентов могут быть соединены между собой, определяя тем самым поток данных в модели.

Многоуровневые модели реализуются путём создания иерархий компонентов, в которой в один компонент может быть вложено несколько компонентов без ограничения по уровням вложенности. Тем самым дочерние компоненты будут являться реализацией родительского. При этом пользовательский интерфейс позволяет абстрагироваться от дочерних компонентов, выдав при этом входы и выходы дочерних компонентов за входы и выходы родительского компонента [2].

III. ПРОГРАММНАЯ РЕАЛИЗАЦИЯ ВИЗУАЛЬНОГО МОДЕЛИРОВАНИЯ

Исполнение модели может происходить либо в системе моделирования, либо отдельно. Для первого случая в самой системе предусмотрена консоль, в которой задаются поля ввода и графики для ввода и вывода данных модели. Поля распределяются по вкладкам, каждая из которых может соответствовать той или иной части моделируемой системы. Каждое из полей имеет свой уникальный в пределах модели идентификатор, на который программист ссылается при реализации программной логики того или иного компонента. При использовании консоли также присутствует возможность протоколирования всех изменений выходных данных на протяжении эксперимента.

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, которая будет содержать необходимые поля ввода и вывода данных, причём их тип и представление не будет ограничиваться средой. 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.

Поскольку наравне с данными эксперимента существуют некоторые внутренние параметры компонента, для них отведена отдельная вкладка окна настройки компонента, где устанавливается соответствие между названием параметра и его значением. Это используется при повторном использовании компонента, позволяя копировать компонент в другую часть системы и менять параметры компонента, не изменяя при этом исходный код.

Непосредственно разработка логики компонентов осуществляется на языке С#: на данный момент доступна разработка с использованием версии С# 4.0. При разработке программного кода создатель модели сначала подключает все необходимые библиотеки и затем разрабатывает логику каждого из компонентов. Глобальные классы и константы описываются в специальном окне. Все данные, описанные здесь, будут доступны из любого компонента. Логика компонента реализуется таким образом, что компонент не имеет представления о модели в целом, выполняя лишь свою роль, получая, обрабатывая и отправляя данные из/на входы компонента.

Разработчику компонента предоставляется некоторая свобода в том плане, что соседние компоненты могут рассматриваться как активные, т. е. посылающие данные рассматриваемому компоненту, либо как пассивные, служащие исключительно для хранения экземпляров сущностей.

IV. ПРОГРАММНЫЙ КОМПЛЕКС ПРОЕКТИРОВАНИЯ ВИЗУАЛЬНОЙ МОДЕЛИ

Исходя из требований к программному комплексу, программа должна разрабатываться для ОС Windows с использованием .NET Framework 4. В связи с этим, наиболее очевидное решение выбора среды программирования – Microsoft Visual Studio .NET Express, которая бесплатна и предоставляет достаточный функционал для реализации программного комплекса, обеспечена возможность разрабатывать отдельно библиотеки и само приложение в одном проекте.

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.

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 соединение для вывода на верхний уровень дерева вложенности. Выбор точек пользователем может быть совершенно произвольным.

Программный комплекс использован для построения модели производственного процесса изготовления силикатных изделий для поиска путей экономии энергетических ресурсов [3, 4].

[1] Маслаков, В. Г. Технология создания визуальных средств имитационного моделирования / В. Г. Маслаков; науч. рук.: А. И. Якимов, К. В. Захарченко // 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.

[2] Якимов, А. И. Технология имитационного моделирования систем управления промышленных предприятий : монография / А. И. Якимов. – Могилев: Белорус.-Рос. ун-т, 2010. – 304 с.: ил.

[3] 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.

[4] Маслаков, В. Г. Имитационная модель процесса изготовления силикатных изделий для поиска путей экономии энергетических ресурсов / В. Г. Маслаков, А. Н. Кунец, Е. А. Якимов // Студенческий вестник, октябрь, 2010. – № 6. [Электрон. ресурс] – Режим доступа: <http://www.bru.mogilev.by/> - Дата доступа 21.10.2010.