

# Метод построения имитационной модели на основе компонентов

Петров Р.В., Альховик С.А.  
Кафедра АСУ, электротехнический факультет  
ГУВПО «Белорусско-Российский университет»  
г. Могилев, Беларусь  
e-mail: ykm@tut.by

**Аннотация** – Рассматривается задача построения имитационной модели сложной системы с использованием компонентного подхода, при котором исходная система декомпозируется на составные элементы со своими алгоритмами моделирования, и взаимодействие моделей этих элементов определяет модель всей системы. В докладе рассматривается структура модели, алгоритм изменения модельного времени и пример построения имитационной модели производственной операции.

**Ключевые слова:** имитационная модель; сложная система; компонентный подход; модельное время.

## I. ВВЕДЕНИЕ

Существует класс сложных систем, к которым относятся, например, производственно-экономические системы, создание имитационных моделей которых является трудоемкой задачей, как на этапе реализации, так и при дальнейшем расширении и уточнении [1]. Использование современных технологий позволяет не только автоматизировать разработку, но и упростить описание имитационной модели, наглядно связать его с реальной системой, что позволит снизить затраты на расширение и уточнение модели.

## II. СТРУКТУРА ИМИТАЦИОННОЙ МОДЕЛИ

Имитационная модель состоит из компонентов связанных в определенную структуру и взаимодействующих между собой. Каждый компонент, при изменении модельного времени в системе, «активизируется» и во время своей «активизации» выполняет некоторый алгоритм (процесс) из предметной области. Компоненты по отношению к декомпозируемому (родительскому) компоненту называются «дочерними компонентами». А дочерние компоненты одного родителя по отношению друг к другу называются «соседние компоненты».

Компонент имеет следующие характеристики.

Параметры – набор некоторых значений, задающий характеристики компонента и определяющие его состояние в процессе работы. Параметры компонента могут задаваться пользователем или другим компонентом как начальные значения, также параметры могут меняться в процессе работы компонента и определять его текущее состояние (отклики).

Интерфейс – компоненты могут взаимодействовать между собой, для этого каждый компонент реализует некоторый набор методов (функций), определяющих его интерфейс. Методы служат для изменения состояния компонента. Компонент может реализовывать несколько интерфейсов (свой и других компонентов).

Состояния – каждый компонент обладает набором состояний, для каждого состояния реализуется свой алгоритм работы. Компонент всегда находится в определенном состоянии и во время «активизации» вызывается алгоритм, связанный с этим состоянием. Алгоритм может менять состояние компонента и в нем

могут вызываться методы других компонентов, а также изменяться значения параметров. Алгоритм компонента определяет момент модельного времени, когда должна произойти следующая активизация компонента.

Дочерние компоненты – у каждого компонента есть список дочерних компонентов, которые могут быть созданы в процессе работы. Дочерние компоненты декомпозируют основной компонент, представляя его в качестве подмодели. Основной компонент может вызывать методы дочерних компонентов и задавать и изменять их параметры для реализации своих собственных методов (интерфейса). При описании списка «Дочерних компонентов» указывается точный тип компонентов, таким образом, структура системы в процессе выполнения детерминирована.

Соседние компоненты – у каждого компонента есть список интерфейсов компонентов, с которыми он может взаимодействовать. Компоненты объединяются на основе списков «Соседних компонентов», тем самым, определяя структуру имитационной модели. При описании списка «Соседних компонентов» указываются необходимые интерфейсы. Таким образом, в качестве «Соседнего компонента» может быть установлен любой компонент реализующий требуемый интерфейс.

Имитационная модель также представляет собой компонент, который может содержать дочерние компоненты, обладает набором параметров (начальных параметров и откликов), в качестве откликов могут быть использованы параметры дочерних компонентов с любого уровня.

## III. АЛГОРИТМ ИЗМЕНЕНИЯ МОДЕЛЬНОГО ВРЕМЕНИ

Выполнение имитационной модели осуществляется путем последовательной активизации компонентов отсортированных по приоритету. Компоненты активизируются рекурсивно, перед активизацией родительского компонента активизируются все его дочерние компоненты. При активизации вызывается метода реализующий алгоритм компонента, в качестве параметра в этот метод передается текущее модельное время, а результатом является интервал модельного времени, через который компонент должен быть активирован заново. В качестве интервала модельного времени, возвращаемого алгоритмом компонента, может быть указано одно из следующих значений: нулевое – означающее, что компоненты должны быть активизированы еще раз без изменения модельного времени (это может понадобиться при смене состояния компонента и необходимости вызова алгоритма заново); флаг «следующий момент времени» – означающее, что изменение модельного времени должно быть рассчитано на основе остальных дочерних компонентов (это может понадобиться, если алгоритму компонента требуется ожидать некоторое событие); число от нуля до бесконечности – максимальное значение интервала следующей активизации

компонента, которое будет учтено при расчете минимального изменения модельного времени.

Особенностью алгоритма активизации компонентов является то, что значение интервала модельного времени, возвращенного компонентом, используется только для расчета общего изменения модельного времени, а активизация всех дочерних компонентов одного родителя всегда происходит «одновременно» при любом изменении модельного времени. Такое поведение алгоритма активизации позволяет компонентам отслеживать наступление нужного события при изменении состояния системы.

На рис. 1 представлена диаграмма последовательности активизации компонентов и изменения модельного времени. Диаграмма отражает процесс выполнения имитационной модели, состоящей из пяти компонентов: модель, K1, K1.1, K1.2 и K2. Числами указана последовательность активации компонентов. Первым активируется компонент «Модель» и определяет время следующей своей активизации. После чего, до наступления этого момента времени, аналогично активизируются дочерние компоненты «Модели» – K1 и K2, а также дочерние компоненты компонента K1 – K1.1 и K1.2. Процесс выполняется рекурсивно.

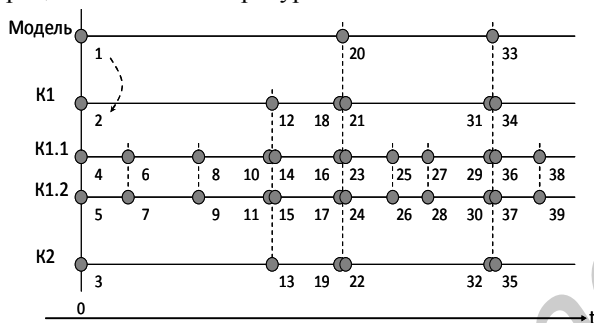


Рис. 1. Диаграмма последовательности активизации компонентов и изменения модельного времени

#### IV. ПРИМЕР ПОСТРОЕНИЯ ИМИТАЦИОННОЙ МОДЕЛИ

Проектирование имитационной модели происходит в два этапа. На первом этапе имитационная модель формализуется в виде текстового описания компонентов, их состояний, параметров, функций и связей между собой, а также в виде графического описания из блоков и стрелок. На втором этапе определяются типы компонентов, функции объединяются в интерфейсы и определяются связи между компонентами на основе интерфейсов.

Ниже представлен пример построения простейшей имитационной модели в терминах компонентов. На рис. 2 изображена схема производственной операции по выточке деталей. Оператор берет заготовку из контейнера, закрепляет ее на станке и начинает обработку заготовки. По окончании обработки готовая деталь автоматически сбрасывается в контейнер с деталями.

На схеме представлено четыре компонента: контейнер с заготовками, оператор, станок, контейнер с деталями. Для взаимодействия между собой одни компоненты реализуют некоторые функции, а другие компоненты эти функции вызывают. В приведенном примере компонент «Контейнер с заготовками» реализует функцию по выдаче заготовки, а «Станок» реализует функцию закрепления детали, эти функции

вызываются компонентом «Оператор» для доставания заготовки из контейнера и закрепления ее на станке. Компонент «Контейнер с деталями» реализует функцию добавления детали, а компонент «Станок» вызывает ее после окончания обработки заготовки. Кроме реализации и вызова функций, компоненты также могут обладать набором состояний и переходить из одного состояния в другое. Например, у компонента «Станок» два состояния «Изготовление детали» и «Ожидание заготовки» в которые он поочередно переходит в процессе выполнения имитационной модели.

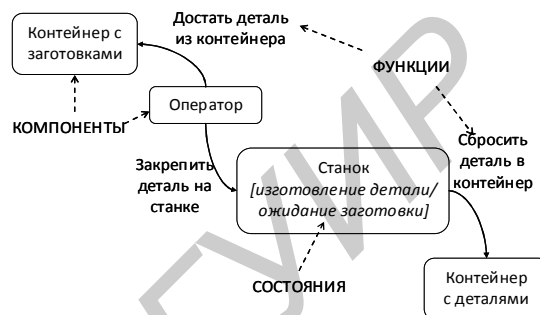


Рис. 2. Схема производственной операции по выточке деталей

Компоненты, объединенные в группы, могут выступать в качестве отдельного компонента более высокого уровня. Например, группа компонентов из предыдущего примера может быть объединена в отдельный компонент «Производственная операция», а несколько таких компонентов могут быть объединены, например, в компонент «Цех».

#### V. ЗАКЛЮЧЕНИЕ

Среда выполнения имитационных моделей реализована на платформе .NET Framework. Программирование выполняется на языке C#, каждый компонент имитационной модели представляет собой .NET класс, унаследованный от базового класса реализующего интерфейс для взаимодействия со средой выполнения. В классе компонента описываются параметры, состояния, функции, алгоритмы состояний, дочерние и соседние компоненты, внешние источники данных в виде полей и методов. Программирование и отладка модели ведется в среде Microsoft Visual Studio с использованием подключаемых модулей расширения, которые на основе рассмотренного выше описания структуры генерируют код модели [2].

[1] Якимов А.И., Альховик С.А. Имитационное моделирование в ERP-системах управления. – Мн.: Бел. наука. - 2005. – 197 с.: ил.

[2] Петров Р.В. Интегрированная среда имитационного моделирования / Р. В. Петров; науч. рук.: С. А. Альховик // 44-я студенческая научно-техническая конференция: материалы конф., редкол.: И. С. Сазонов (гл. ред.) [и др.]. – Могилев: Бел.-Рос. ун-т, 2008. – С. 156-158.