

---

**МОДЕЛЬНЫЙ СИНТЕЗ И МОДЕЛЬНО-ОРИЕНТИРОВАННОЕ  
ПРОГРАММИРОВАНИЕ, КАК ТЕХНОЛОГИЯ РЕАЛИЗАЦИИ ИМИТАЦИОННЫХ  
МОДЕЛЕЙ СЛОЖНЫХ МНОГОКОМПОНЕНТНЫХ СИСТЕМ, С ОРИЕНТАЦИЕЙ  
НА ПАРАЛЛЕЛЬНЫЕ И РАСПРЕДЕЛЕННЫЕ ВЫЧИСЛЕНИЯ****Ю.И. Бродский (Москва)**

Модельный анализ как способ описания и синтеза имитационных моделей сложных многокомпонентных систем развивался в отделе Имитационных систем ВЦ РАН с конца 80-х гг. Основные его идеи и методы изложены в работах [1-5], однако сам термин «модельный анализ» впервые вводится в работе [1]. В основе модельного анализа лежит понятие модели-компоненты. С содержательной точки зрения, модель-компонента подобна объекту объектного анализа, но снабженному не только методами, способными делать что-то полезное, если их вызовут, но и неким аналогом операционной системы, всегда готовой давать стандартные ответы на стандартные запросы внутренней и внешней среды модели.

С точки зрения теории родов структур Н. Бурбаки [6], модель-компонента есть математический объект, базисным множеством которого является совокупность множеств внутренних и внешних характеристик модели, методов (того, что модель умеет делать) и событий (того, на что модель должна уметь реагировать). На базисном множестве вводится структура рода «модель-компонента», которая обладает двумя замечательными свойствами:

1. Род структуры «модель-компонента» позволяет по нему стандартным и однозначным образом организовать вычислительный процесс моделирования для любого математического объекта, снабженного такой структурой. Это означает возможность создания универсальной программы, способной запустить на выполнение любую имитационную модель, если та является математическим объектом, снабженным структурой рода «модель-компонента».
2. Вообще говоря, если рассмотреть два произвольных математических объекта, снабженных структурой одного рода, то распространение этой структуры на объединение их базисных множеств возможно далеко не всегда. Тем не менее для рода структуры «модель-компонента» подобное распространение общей структуры компонент на объединение их базисных множеств или возможно (если подмножества характеристик их базовых множеств не имеют попарных пересечений), или возможно с некоторыми оговорками (например, при условии пополнения исходного набора объектов-компонент еще некоторым количеством объектов, снабженных той же структурой).

Опишем устройство модели-компоненты на неформальном уровне.

**Характеристики**

Модель-компонента, подобно объекту объектного анализа, имеет характеристики. Эти характеристики мы будем разбивать на внутренние и внешние. Внутренние характеристики – это то, что модель-компонента моделирует, внешние – это то, что она знает о внешнем мире.

**Процессы**

Функциональность модели-компоненты удобно структурировать следующим образом. Считается, что модель-компонента реализует один или несколько параллельно выполняющихся процессов. Процесс состоит в последовательном чередовании элементов – алгоритмически элементарных методов. Если какой-либо процесс какую-то часть модель-

ного времени не выполняется – удобно считать, что в это время он выполняет «пустой» элемент, не меняющий никаких характеристик модели-компоненты.

### Элементы

Элементарные, алгоритмически однородные методы, реализующие функциональности модели-компоненты (т. е. то, что компонента умеет делать, получение на основании значений некоторых внутренних и внешних характеристик модели-компоненты, новых значений некоторых ее внутренних характеристик).

По отношению к модельному времени некоторые элементы выполняются мгновенно, это сосредоточенные или быстрые элементы. Быстрыми элементами можно моделировать дискретные характеристики системы.

Выполнение других элементов занимает определенное время. Если при этом элемент для любого промежутка времени  $\Delta t$ , не превосходящего стандартный шаг моделирования  $\Delta t$  выдает некий осмысленный результат, такой элемент называется распределенным или медленным. Распределенные элементы – естественное средство вычисления непрерывных характеристик модели.

Частью предлагаемой концепции является жесткая дисциплина работы методов с внутренними характеристиками модели: каждый метод имеет право изменять только «свои» характеристики. Эта дисциплина основана на принятии предположения о детерминированности и однозначности имитационных вычислений. В рамках предлагаемой концепции, конфликт доступа, возникающий, когда методы  $A$  и  $B$  вычисляют одну и ту же характеристику  $X = X_A$  и  $X = X_B$ , может быть разрешен, например, введением метода  $C$ , который, получая на входе в качестве параметров  $X_A$  и  $X_B$ , вычисляет на их основании искомую характеристику  $X$ , устраняя тем самым не только конфликт доступа к ней, но и, очевидно, имевшую место неоднозначность вычисления упомянутой характеристики. Принятая дисциплина доступа к характеристикам позволяет вызывать параллельно те методы, которые в модельном времени выполняются одновременно.

Наконец, последнее, что следует заметить относительно элементов. Как и всякий метод, каждый элемент имеет входные и возвращаемые параметры. Концепция моделирования предполагает возможность распределенного вычисления элементов, т. е. элемент, вообще говоря, может быть найден разработчиком модели-компоненты на просторах Интернета, поэтому его параметры таковы, какими их сделал его автор, и скорее всего, никак не согласованы с характеристиками модели-компоненты, которые придумал ее разработчик. Поэтому при описании компоненты обязательно должно быть уделено внимание описанию коммутаций входных параметров элементов с внутренними и внешними характеристиками компоненты и выходных параметров элементов – с ее внутренними характеристиками.

### События

Содержательно, события – это то, что нельзя пропустить при моделировании динамики системы – точки синхронизации различных ее функциональностей, представляемых процессами. Точки, когда получены такие значения характеристик модели и внешних характеристик, на которые обязаны отреагировать некоторые процессы модели-компоненты.

Формально событие – функция значений внутренних и внешних характеристик в начале шага моделирования. С точки зрения организации имитационных вычислений, событие – это метод, входными параметрами которого является подмножество внутренних и внешних характеристик модели-компоненты, а выходной параметр один – прогнозируемое время до наступления этого события. Если это прогнозируемое время равно нулю – значит, событие уже наступило. События управляют чередованием элементов в процессе.

### Выполнение модели-компоненты

Модель-компонента – элементарная, но тем не менее полноправная модель сложной системы. Поэтому ее можно запустить на выполнение имитационного эксперимента. Конечно, если имеются начальные данные и способ нахождения внешних характеристик модели-компоненты в моменты событий.

Правила запуска модели-компоненты на выполнение следующие. Во-первых, задается стандартный шаг моделирования  $\Delta t$ . Во-вторых, считается, что в начале шага моделирования известны текущие элементы всех процессов и все внутренние и внешние характеристики модели. Далее:

1. Вычисляются события связанные с текущими элементами процессов. **Все эти события можно вычислять параллельно.** Если есть наступившие события, проверяется нет ли переходов к быстрым (сосредоточенным) элементам, если они есть – **параллельно** выполняются быстрые элементы (они становятся текущими), затем возврат к началу п.1; если нет переходов к быстрым элементам – совершаются переходы к новым медленным (распределенным) элементам, затем возврат к началу п.1.
2. Если нет наступивших событий – из всех прогнозов событий выбирается ближайший  $\Delta \tau$ .
3. Если стандартный шаг моделирования не превосходит прогнозируемого времени до ближайшего события,  $\Delta t \leq \Delta \tau$  – **параллельно** моделируем текущие распределенные элементы со стандартным шагом  $\Delta t$ . В противном случае – **параллельно** моделируем их с шагом времени до ближайшего спрогнозированного события  $\Delta \tau$ .
4. Возвращаемся к началу п.1.

Все события и элементы есть функции в математическом смысле и в смысле парадигмы функционального программирования, т.е. не имеют состояний и побочных эффектов, что может дать при их вычислении дополнительные возможности распараллеливания.

В связи с приведенными правилами выполнения модели-компоненты могут возникнуть вопросы: не могут ли выполнение быстрых элементов в п. 1, а также уменьшение шага времени в п. 3 привести к зацикливанию программы за счет возникновения точек накопления системных событий. Полную гарантию можно дать лишь для кусочно-гладких непрерывных слева систем.

Еще заметим, что раз мы требовали от элементов однозначности имитационных вычислений и сумели этого добиться – на выполнение все одновременно выполняющиеся в модельном времени элементы можно запускать асинхронно, т.е. загружать ими имеющиеся ядра процессора или же доступные распределенные компьютеры. Если же этого почему-то добиться не удалось – у системы поддержки выполнения модели есть полная информация о том, кто что меняет (собственно это она должна будет по выполнении элементов обновить соответствующие данные в базе) – и она обязана выдать соответствующую диагностику ошибки времени выполнения.

### Модели-комплексы

Модели-компоненты могут объединяться в модели-комплексы, при этом (необязательно) может оказаться, что некоторые компоненты явно моделируют внешние характеристики некоторых других компонент. Для того чтобы полностью описать модель-комплекс, достаточно указать:

1. Какие модели-компоненты и в каком количестве экземпляров в него входят.
2. Коммутацию моделей-компонент внутри модели-комплекса, если она имеет место, т. е. какие внутренние характеристики каких моделей-компонент являются какими внешними характеристиками и каких именно компонент модели-комплекса.

При объединении компонент в комплекс, следует иметь в виду, что однозначность вычислительного процесса может быть потеряна. Так может произойти, если по каким-то причинам, несколько компонент вычисляют одну и ту же характеристику моделируемого явления. В таком случае можно ввести в модель-комплекс новую компоненту, которая в качестве внешних характеристик получает весь многозначный набор значений упомянутой характеристики, а в качестве внутренней характеристики каким-то образом вычисляет единственное ее значение.

#### **Комплекс как модель-компонента**

Модель-комплекс, состоящий из многих моделей-компонент, во вне может проявляться в качестве единой модели-компоненты.

Введем следующую операцию объединения компонент комплекса:

1. Внутренними характеристиками комплекса объявляется объединение внутренних характеристик всех его компонент.
2. Процессами комплекса объявляется объединение всех процессов его компонент.
3. Методами комплекса объявляется объединение всех методов его компонент.
4. Событиями комплекса объявляется объединение всех событий его компонент.
5. Внешними характеристиками комплекса объявляется объединение всех внешних характеристик его компонент, из которого исключаются все те характеристики, которые моделируются явно какими-либо компонентами. Операция объединения превращает модель-комплекс в модель-компоненту.

#### **Заключение**

Предложена организация имитационных вычислений, ориентированная на модели с кусочно-гладкими траекториями, как инварианта относительно объединения моделей-компонент в модели-комплексы, позволяющая полностью решить задачу описания и синтеза имитационных моделей сложных многокомпонентных систем.

Все это позволяет формально определить класс имитационных моделей сложных многокомпонентных систем как семейство родов структур «модель-компонента» в смысле Н. Бурбаки, и на основе такого определения построить новые концепции формального описания, синтеза и реализации имитационных моделей – модельный синтез и модельно-ориентированное программирование, являющиеся альтернативой повсеместно используемых объектного анализа и объектно-ориентированного программирования, прежде всего для задач имитационного моделирования сложных многокомпонентных систем.

Концепция модельного синтеза минималистична в отношении базовых понятий и представлений: в ней всего одно основное понятие – модель-компонента, и одно вспомогательное – модель-комплекс, который путем несложных операций синтеза, описанных выше, в конечном итоге также превращается в модель-компоненту.

Концепции модельного синтеза и модельно-ориентированного программирования полностью исключают из проекта создания имитационной модели наиболее трудоемкое в проектировании, реализации и отладке императивное программирование, ограничиваясь лишь программированием декларативным и функциональным [5]. Кроме того, модельно-ориентированное программирование производит код высокой степени параллельности.

Разработанная концепция моделирования сложных систем была реализована на практике в виде ряда имитационных систем, реализованных под влиянием модельно-ориентированной парадигмы программирования, инструментальной системы имитационного моделирования MISS [3], а также в виде программного обеспечения макета рабочей станции пиринговой системы распределенного имитационного моделирования [2]. В настоящее время в отделе Имитационных систем и исследования операций ВЦ РАН ведутся

---

работы по реализации системы модельно-ориентированного программирования для высокопроизводительных вычислительных систем.

Изложенная концепция модельного синтеза применима в первую очередь для синтеза моделей сложных многокомпонентных систем. Однако можно надеяться, что подобный подход применим и для разработки сложных программных систем, организация которых атомистична и укладывается в описанные во втором разделе требования гипотезы о замкнутости, а также требования однозначности и детерминированности вычислений.

*Работа выполнена при финансовой поддержке РФФИ в рамках научного проекта №13-01-00499-а, и РГНФ в рамках научного проекта №12-06-00932-а.*

### Литература

- 1. Бродский Ю.И.** Модельный синтез и модельно-ориентированное программирование. – М.: ВЦ РАН, 2013. – 140 с.
- 2. Бродский Ю.И.** Распределенное имитационное моделирование сложных систем. – М.: ВЦ РАН, 2010. – 156 с.
- 3. Бродский Ю.И., Лебедев В.Ю.** Инструментальная система имитации MISS. – М.: ВЦ АН СССР, 1991. – 180 с.
- 4. Бродский Ю.И., Павловский Ю.Н.** Разработка инструментальной системы распределенного имитационного моделирования // Информационные технологии и вычислительные системы. – 2009. – № 4. – С. 9-21.
- 5. Бродский Ю.И., Мягков А.Н.** Декларативное и императивное программирование в имитационном моделировании сложных многокомпонентных систем // Вестник МГТУ им. Н.Э. Баумана. Сер. Естественные науки. Спец. выпуск № 4 «Математическое моделирование». – 2012. – С.178–187.
- 6. Бурбаки Н.** Теория множеств. – М.: Мир, 1965. – 456 с.