

## ИНФОРМАТИКА И ПРОГРАММИРОВАНИЕ

УДК 004.415.2

**К.Ю. Войтиков, А.Н. Моисеев, П.Н. Тумаев**

### КОМПОНЕНТНАЯ МОДЕЛЬ РАСПРЕДЕЛЕННОЙ ОБЪЕКТНО-ОРИЕНТИРОВАННОЙ СИСТЕМЫ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ<sup>1</sup>

В работе рассматриваются вопросы построения объектной распределенной системы моделирования процессов массового обслуживания, в частности, предложена компонентная модель системы и техника организации распределенных вычислений.

**Ключевые слова:** *объектно-ориентированное проектирование, распределенные вычисления, имитационное моделирование.*

На сегодняшний день на рынке программного обеспечения существует определенное число систем, предназначенных для моделирования процессов массового обслуживания. Большая часть таких систем ориентирована на решение практических задач в конкретной предметной области (транспорт, телекоммуникации и т.п.), к тому же практически все эти системы реализуют определенные модели имитационного моделирования, не предусматривая подключения других возможных подходов. В качестве примера программы, удачно сочетающей решение практических задач в конкретных областях с возможностью реализации того или иного теоретического подхода, можно привести систему AnyLogic компании XJ Technologies [1]. Пожалуй, это одна из лучших систем в своем классе. Однако и эта программа обладает рядом недостатков, наиболее существенным из которых является необходимость использовать для проведения вычислений производственные мощности настольного (клиентского) компьютера. В связи с практическим отсутствием средств, решающих задачи моделирования с возможностью распределенных вычислений (в том числе – и по сети Интернет), актуальной является задача создания соответствующего программного обеспечения.

#### 1. Базовые компоненты системы

В качестве дополнительного требования к разрабатываемой системе была выдвинута возможность производить независимо от имитационного и математическое моделирование систем массового обслуживания (где это возможно) и представлять сравнительные результаты расчетов на примере конкретных систем.

---

<sup>1</sup> Работа выполнена в рамках аналитической ведомственной целевой программы «Развитие научного потенциала высшей школы (2009 – 2010 годы)», проект № 4761.

В качестве базовой парадигмы программирования выбран объектно-ориентированный подход. В процессе разработки была получена компонентная модель системы, ее общий вид представлен на рис. 1.

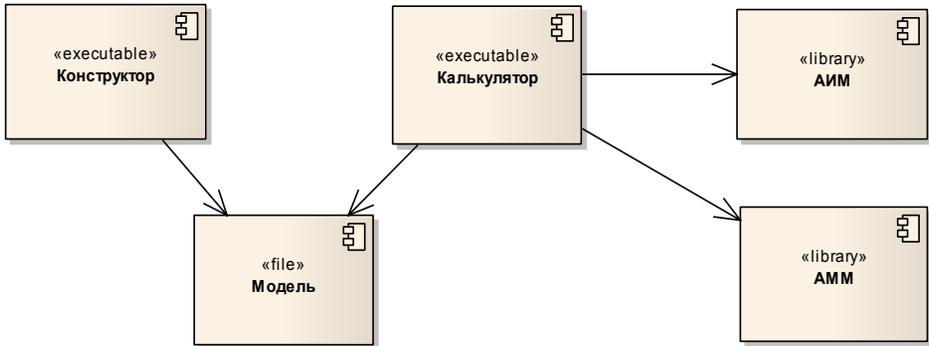


Рис. 1. Модель основных компонентов объектной распределенной системы моделирования

В качестве отдельного компонента, например в виде инструментальной библиотеки, предлагается выделить алгоритмы имитационного моделирования (АИМ). Данный компонент реализует набор механизмов для построения имитационной модели систем массового обслуживания, а также базовые инструменты имитационного моделирования.

Для выполнения расчетов, не нуждающихся в имитационном моделировании, расчетов, имеющих аналитические или численные решения, предлагается использовать компонент, отвечающий за алгоритмы математического моделирования (АММ). Кроме того, совместное использование этих компонентов позволит не только провести численные расчеты и выполнить имитационное моделирование, а при необходимости и сравнить полученные результаты.

Калькулятор (Calculator) – «ядро» системы, производит чтение и разбор данных, описанных в файле Модель, и обращается к компонентам АИМ (AIM) и АММ (AMM) для выполнения соответствующих расчетов;

Модель (Model) – компонент типа «файл», полностью описывающий постановку задачи моделирования и все необходимые параметры. Для этого компонента выбрана объектная модель внутренней структуры;

Конструктор (Constructor) – программа, предоставляющая возможность создания и редактирования моделей в удобном для пользователя виде.

Исходя из предполагаемой сложности, а главное длительности выполняемых вычислений, возникает задача разработки архитектуры, позволяющей использовать для работы системы несколько компьютеров одновременно, причем таким образом, что все участвующие в работе системы компьютеры должны одновременно использоваться для выполнения одного текущего задания системы, то есть расчетов по одной и той же задаче.

Для построения системы воспользуемся принципом распределённых вычислений (GRID, MPP[2]). Это частный случай параллельных вычислений, при котором используется большое количество самостоятельных компьютеров, а разделение расчетов осуществляется путем разбиения основной задачи на ряд подзадач.

Отсюда следует, что принципиально такая архитектура может использоваться лишь для тех моделей, имитацию которых можно разбить на независимые части без потери адекватности имитации.

## 2. Организация распределенных вычислений

В работе системы должны участвовать три типа узлов:

- Клиент (Client) – компьютер клиента системы, который составляет для нее задание;
- Удаленный Калькулятор (Remote Calculator) – узел, использующийся непосредственно для расчетов;
- Сервер (Server) – узел, использующийся для приема заданий от Клиентов, их хранения, координации сети Удаленных Калькуляторов, аккумуляцию и синхронизацию результатов расчетов.

Как видно из модели компонентов, представленной на рис. 1, непосредственно в процессе имитации и вычислений обязательно участвуют три ее компонента – Calculator, AIM и AMM. Вдобавок экземпляры компонента Model передаются компоненту Calculator для каждого конкретного задания.

Исходя из этого, компоненты основной модели системы будут распределены по типам узлов следующим образом (рис. 2).

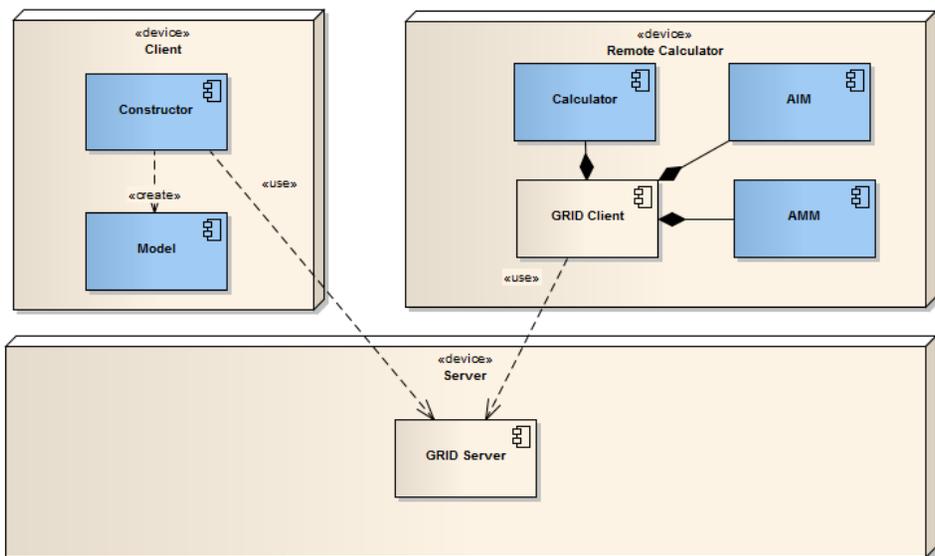


Рис. 2. Диаграмма развертывания компонентов системы

Наряду с основными компонентами в системе появляются два новых компонента: **GRID Client** – компонент, агрегирующий в себя компоненты, участвующие в расчетах, и обеспечивающий их связь с Сервером; и **GRID Server** – компонент, осуществляющий разделение заданий на сегменты, обеспечивающий координацию Удаленных Калькуляторов и Клиентов, хранящий задания и результаты имитаций.

Стоит заметить, что узел Клиент не обязательно содержит компонент Конструктор. В общем случае достаточно передать компоненту GRID Server файл модели и параметры задания для нее, для чего Конструктор не является необходимым. Для передачи этих параметров будет использоваться специальный интерфейс Сервера, названный Интерфейсом Заданий.

Компонент Model фактически присутствует на всех трех узлах системы, так как создается пользователем на узле Клиент, передается и хранится вместе с заданием на узле Сервер, а также передается и обрабатывается на узле Удаленный Калькулятор.

Одним из ключевых свойств такой архитектуры является пассивность компонента GRID Server, который, по сути, является хранилищем данных и правил доступа к ним. Таким образом, взаимодействие компонентов системы, находящихся на узлах Клиент и Удаленный Калькулятор осуществляется через изменение данных, хранящихся в компоненте GRID Server. Так клиент системы создает новое задание и сохраняет его на Сервере. Множество Удаленных Калькуляторов в определенные промежутки времени соединяются с Сервером и, если он содержит новые задания, загружают их и начинают процесс имитации, наполняя Сервер результатами. По окончании моделирования всеми Удаленными Калькуляторами, клиент системы вновь подключается к Серверу и получает необходимые данные о результатах моделирования.

Из этого следует, что компонент GRID Server должен предоставлять два интерфейса: Интерфейс Заданий для создания заданий и получения результатов клиентом и интерфейс GRID Клиентов для подключения Удаленных Калькуляторов.

### 3. Применение WCF для организации распределенных вычислений

Для связи компонентов системы между узлами была выбрана технология Microsoft Windows Communication Foundation (WCF). WCF – это унифицированная модель программирования распределенных приложений на платформе Microsoft. Она инкорпорирует предшествующие технологии – ASMX, .NET Remoting, DCOM и MSMQ – и предоставляет расширяемый API, отвечающий разнообразным требованиям, которые возникают при создании распределенных систем. WCF – это каркас для работы с web-службами на основе XML, который совместим со многими другими технологиями [3]. WCF предоставляет возможность конструировать логику взаимодействия независимо от протоколов, форматов и способов передачи данных, позволяет динамически переключаться между ними, а также предоставляет готовые механизмы обеспечения безопасности исполнения кода.

Рассмотрим более подробно структуру компонентов GRID Server и GRID Client.

Центральным звеном компонента GRID Server является компонент Controller, отвечающий за обработку всех событий внутри компонента и реализующий бизнес-логику обращения к данным.

Как было сказано выше, компонент GRID Server предоставляет два интерфейса: Интерфейс Заданий и Интерфейс GRID Клиентов. Интерфейс GRID Клиентов реализуется компонентом Service, который является WCF-службой и обеспечивает связь компонентов GRID Client и GRID Server вне зависимости от способа физического соединения содержащих их узлов и доступных протоколов передачи данных.

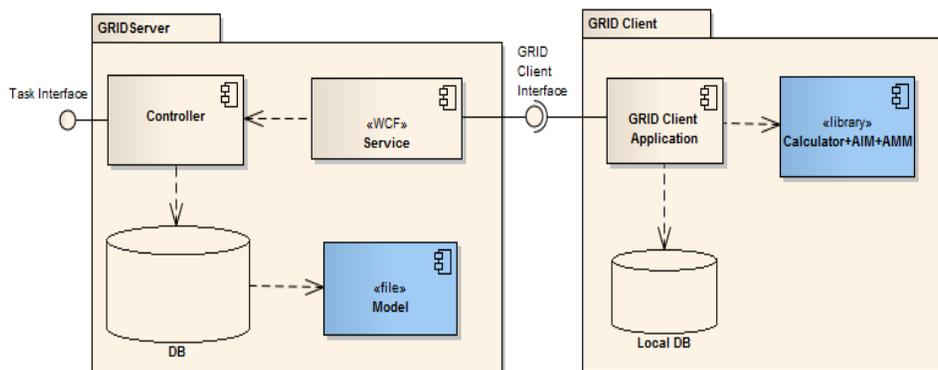


Рис. 3. Структура компонентов GRID Server и GRID Client

Компонент DB является базой данных, хранящей информацию о заданиях системы, результатах проведенных ранее имитаций, а также о существующих GRID Клиентах.

Компонент Model – компонент из основной модели системы. Отдельные экземпляры компонента передаются на Сервер Клиентом и привязываются к конкретному заданию.

Компонент Grid Client Application – центральное звено GRID Клиента. Компонент выполняет запросы к WCF-службе Сервера для получения новых заданий и отправки результатов имитации. Иницирует имитацию и выполнение заданий компонентом Calculator+AIM+AMM.

Компонент LocalDB является локальной базой данных или другой формой хранения данных для локального хранения заданий и результатов между сеансами связи с GRID Сервером.

Стоит отметить, что гибкость интеграции и разделения компонентов вычислительной системы достигается благодаря применению проектных решений Фасад [4] и Шлюз [5]. Применение этих проектных решений позволяет избавиться от возможной зависимости систем друг от друга, выделить их в независимые программные модули, что, в свою очередь, способствует повышению степени их повторной используемости.

### Заключение

Итак, в работе представлена модель распределенной системы моделирования процессов систем массового обслуживания, работоспособность которой в достаточной степени не зависит от физического размещения компьютеров и пользователей, участвующих в работе системы, а также доступных каналов связи между ними. Это делает доступным для привлечения к работе большее количество вычислительных ресурсов, а саму систему – доступной для большего количества клиентов.

### ЛИТЕРАТУРА

1. *XJ Technologies* (Официальный русскоязычный сайт) [Электронный ресурс]. URL: <http://www.xjtek.ru>
2. *Топорков В. В.* Модели распределенных вычислений. М.: ФИЗМАТЛИТ, 2004. 320 с.

3. Резник С., Крейн Р., Боуэн К. Основы Windows Communication Foundation для .NET Framework 3.5: пер с англ. М.: ДМК Пресс, 2008. 480 с.
4. Гамма Э., Хелм Р., Джонсон Р., Влиссидес Дж. Приемы объектно-ориентированного проектирования: паттерны проектирования. СПб.: Питер, 2001. 366 с.
5. Фаулер М. Архитектура корпоративных программных приложений: пер. с англ. М.: Изд. дом «Вильямс», 2004. 544 с.

Статья представлена оргкомитетом VIII Всероссийской научно-практической конференции с международным участием «Информационные технологии и математическое моделирование».

*Войтиков Константин Юрьевич*

Анжеро-Судженский филиал Кемеровского государственного университета

*Моисеев Александр Николаевич*

Томский государственный университет

*Тумаев Павел Николаевич*

Анжеро-Судженский филиал Кемеровского государственного университета

E-mail: kost\_v@ngs.ru; amoiseev@ngs.ru; ptumaev@yandex.ru

Поступила в редакцию 21 января 2010 г.