

УДК 004.94

А.Н. Моисеев, М.В. Сняжков

РАЗРАБОТКА ОБЪЕКТНО-ОРИЕНТИРОВАННОЙ МОДЕЛИ СИСТЕМЫ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ ПРОЦЕССОВ МАССОВОГО ОБСЛУЖИВАНИЯ¹

Статья посвящена разработке объектно-ориентированной модели системы имитационного моделирования процессов массового обслуживания. Реализуемая имитационная модель системы массового обслуживания основана на дискретно-событийном подходе к моделированию. Проектируемая программная система предоставит возможность исследовать характеристики систем массового обслуживания различных конфигураций.

Ключевые слова: имитационное моделирование, система массового обслуживания, объектно-ориентированное проектирование.

Задачи организации массового обслуживания возникают во многих областях практической деятельности. Системы массового обслуживания (СМО) представляют собой системы специального вида, реализующие многократное выполнение однотипных задач.

Система массового обслуживания полностью описывается заданием: входящего потока, дисциплины очереди, порядка обслуживания [1 – 2].

Схематически систему массового обслуживания можно изобразить в следующем виде (рис. 1):

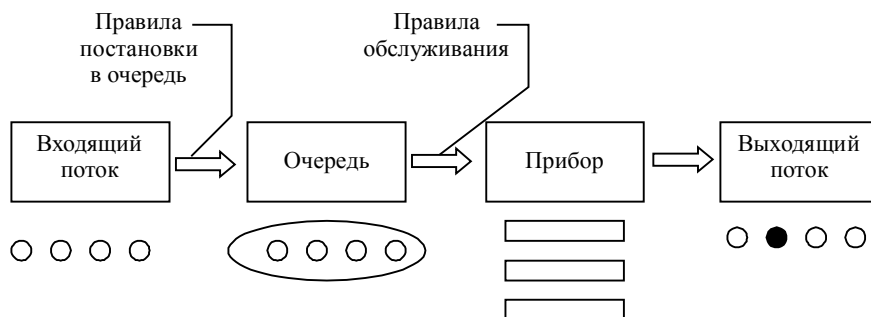


Рис. 1. Система массового обслуживания

Одним из методов исследования сложных систем, в том числе и систем массового обслуживания (СМО), является метод имитационного моделирования, когда модель (как следует из названия вида моделирования) имитирует работу реальной системы, т.е. модель воспроизводит процесс функционирования реальной системы во времени [3].

Целью данной работы является разработка объектно-ориентированной модели системы имитационного моделирования процессов массового обслуживания.

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

Стоит отметить, что данная система является частью объектно-ориентированной распределенной системы моделирования процессов массового обслуживания [4] и представляет собой блок алгоритмов имитационного моделирования (АИМ).

1. Модельное время и события системы

Процесс функционирования системы — это последовательная смена состояний системы во времени.

Процесс перехода системы из одного состояния в другое называется событием системы.

Очевидно, что основными событиями для рассматриваемой нами системы будут:

- Поступление заявки;
- Обращение заявки из источника повторных вызовов (ИПВ);
- Завершение обслуживания заявки на обслуживающем устройстве;
- Завершение моделирования.

Процесс поступления заявки предполагает создание заявки и ее поступление в систему. Затем происходит проверка «занятости прибора», если прибор свободен, то заявка встает на обслуживание, в противном случае добавляется в очередь. При этом процесс постановки в очередь зависит от ограничения длины очереди: если длина очереди достигла предела, то заявка покидает систему.

В процессе моделирования системы таймер модельного времени, безусловно, должен постоянно корректироваться в соответствии с теми основными событиями, которые возникают в реальной системе. При этом существуют два способа (два принципа) коррекции значения таймера модельного времени:

- 1) коррекция с постоянным шагом;
- 2) коррекция с переменным шагом.

В случае коррекции с постоянным шагом модельное время увеличивается на величину Δh , выбираемого перед началом моделирования. При этом постоянно проверяется, не произошло ли событие на интервале $[h, h+\Delta h]$. Очевидно, что при слишком большом значении Δh , результаты моделирования будут недостоверными, тогда как выбор слишком маленького значения Δh приведет к увеличению затрат на моделирование.

При коррекции таймера модельного времени с переменным шагом значения таймера должны непосредственно совпадать с моментами возникновения событий в системе, поэтому на каждой итерации значение таймера сдвигается к моменту очередного события.

В качестве способа коррекции таймера модельного времени будем использовать способ коррекции с переменным шагом в силу очевидных достоинств, присущих данному способу.

Так как в разрабатываемой системе определено 4 события, то таймер модельного времени ($T_{\text{мод}}$), корректируемый в основном имитационном цикле, будет сдвигаться в сторону ближайшего наступившего события. То есть для моментов наступления основных событий:

- $T_{\text{пост}}$ — момент времени поступления заявки в систему
- $T_{\text{ИПВ}}$ — момент времени обращения заявки из ИПВ;
- $T_{\text{заверш.обслуж}}$ — момент времени завершения обслуживания заявки на обслуживающем устройстве;
- $T_{\text{заверш.модел}}$ — момент времени завершения моделирования.

Коррекция модельного времени определяется как

$$T_{\text{мод}} = \min(T_{\text{пост}}, T_{\text{ИПВ}}, T_{\text{заверш.обслуж}}, T_{\text{заверш.модел}}).$$

2. Объектная модель системы имитационного моделирования

Имитационная модель системы массового обслуживания разрабатывается на основе дискретно-событийного подхода к моделированию, предлагающего абстрагироваться от непрерывной природы событий и рассматривать только основные события моделируемой системы [5].

Таким образом, зная основные элементы СМО, логику функционирования и принципы взаимодействия элементов системы, рассмотрим статическую модель имитационной модели системы массового обслуживания (ИМСМО).

Класс *QueueSimulationModel* – центральное звено системы, представляет собой имитационную модель. Создает все основные объекты системы. Назначает порядок прибытия и правила обслуживания заявок. Метод *RunModel* запускает основной имитационный цикл.

В имитационном цикле источники заявок (экземпляры класса *Source*) создают заявки в случайные моменты, определяемые параметрами распределения, формируя входящий поток заявок. Заявка (*Advert*), поступившая в систему, отправляется на прибор (*ServiceDevice*), при этом моменты времени прибытия заявки и ее обращения на прибор совпадают. Если все приборы заняты, то заявка отправляется в хранилище (*Storage*). Хранилище может представлять собой очередь или источник повторных вызовов (*Recall*).

Заявки, находясь в очереди, ждут своего обслуживания на приборе, при этом прибор, освободившись, получает очередную заявку из очереди в соответствии с установленной для конкретной очереди дисциплиной обслуживания элементов (LIFO, FIFO и т.д.). Заявки, находящиеся в ИПВ, вновь обращаются к прибору с попыткой повторного обслуживания через случайные интервалы времени.

Необходимо отметить, что в системе может находиться любое число источников заявок, хранилищ, обслуживающих устройств. При этом порядок взаимодействия элементов системы определяется параметрами имитационной модели и устанавливается объектом *QueueSimulationModel* (рис. 2).

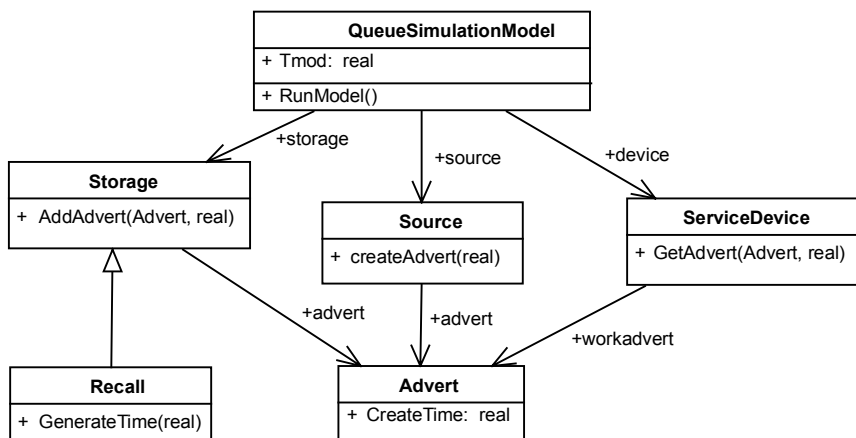


Рис. 2. Основные классы системы

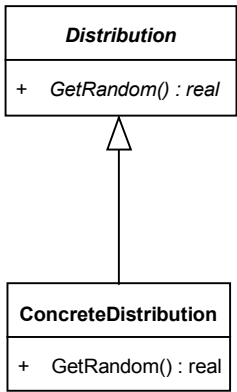


Рис. 3. Класс *Distribution*

Случайные величины, определяющие моменты создания заявок, время обслуживания на приборе или интервал обращения из ИПВ, распределены по определенным законам распределения. За генерацию случайных величин отвечает класс *Distribution* (рис. 3) и его потомки, генерирующие случайные величины, распределенные по конкретному закону распределения.

На рис. 4 отображены взаимодействия объектов, упорядоченные по времени их появления, представляющие процедуру создания заявки, ее постановку в очередь и обслуживание на приборе.

Источник заявок генерирует заявку (метод *createAdvert(real)*) в момент времени, который представляет собой случайную величину, описанную заданным законом распределения (метод *GetRandom():real*). Созданная заявка, в случае занятости прибора, помещается в очередь (метод *AddAdvert(Advert, real)*) и после освобождения прибора извлекается из очереди (*GetFirst():Advert*) и встает на обслуживание (*GetAdvert(Advert, real)*).

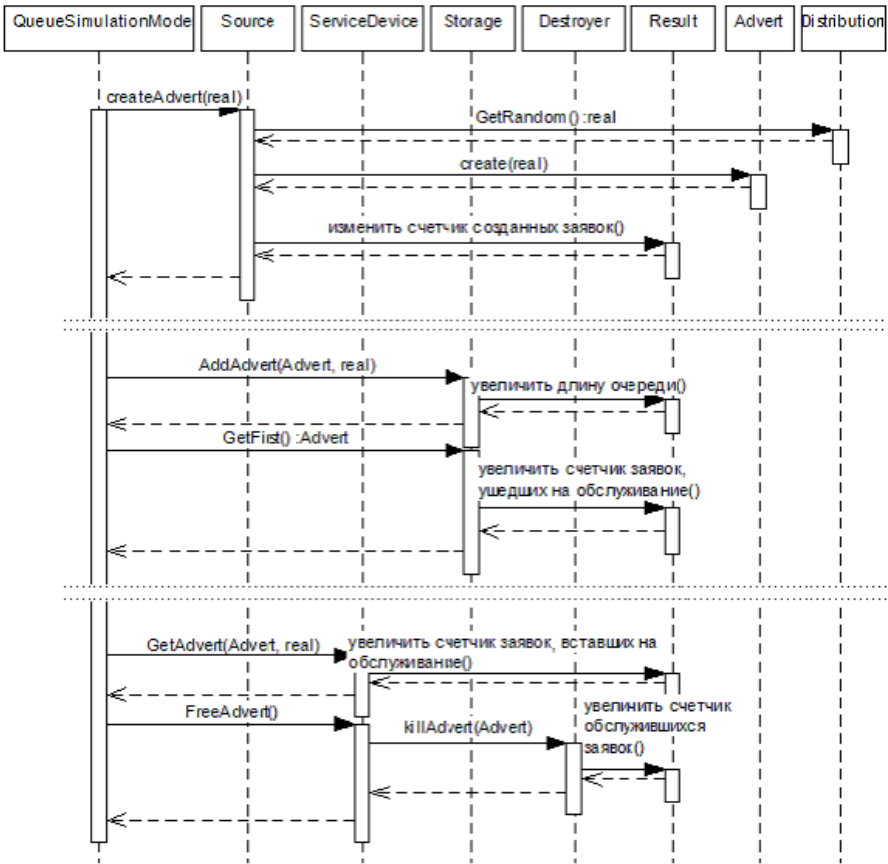


Рис. 4. Взаимодействие объектов системы

Класс *Result* (рис. 4) отвечает за сбор результатов функционирования системы. Элементы системы, которые являются основными инициаторами событий, такие, как очередь, источник заявок, обслуживающее устройство, уничтожитель заявок (*Destroyer*), регистрируют у себя объект *Result* и при изменении своего состояния сообщают ему о соответствующих изменениях. Для реализации такой конструкции целесообразно использовать паттерн объектно-ориентированного проектирования «Наблюдатель» [6].

Заключение

В работе предложен подход к разработке объектно-ориентированной системы имитационного моделирования для решения описанных задач моделирования в общем виде, позволяющий изучать системы массового обслуживания с произвольным числом источников заявок, обслуживающих устройств и очередей.

ЛИТЕРАТУРА

1. Назаров А.А., Терпугов А.Ф. Теория массового обслуживания. Томск: Изд-во НТЛ, 2005. С. 228.
2. Хинчин А.Я. Работы по математической теории массового обслуживания. М.: Физматгиз, 1963. С. 236.
3. Кобелев Н.Б. Основы имитационного моделирования сложных экономических систем. М.: Дело, 2003. 333 с.
4. Войтиков К.Ю. Моисеев А.Н. Модель компонентов объектной распределенной системы моделирования процессов массового обслуживания // ИТММ. 2009. С. 122 – 124.
5. Змеев О.А. Пристupa А.В. Разработка объектно-ориентированного программного комплекса имитационного моделирования систем массового обслуживания // Вестник ТГУ. 2004. № 248. С. 171 – 173.
6. Гамма Э., Хелм Р., Джонсон Р., Влессидес Дж. Приемы объектно-ориентированного проектирования. Паттерны проектирования. СПб.: Питер, 2001. С. 368.

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

Моисеев Александр Николаевич

Томский государственный университет

Синяков Михаил Викторович

Анжеро-Судженский филиал Кемеровского государственного университета

E-mail: amoiseev@ngs.ru; protocol@ngs.ru

Поступила в редакцию 21 января 2010 г.