

**МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
им. М.В. Ломоносова**
ФАКУЛЬТЕТ ВЫЧИСЛИТЕЛЬНОЙ МАТЕМАТИКИ и КИБИРНЕТИКИ

**IV Международная
научно-практическая конференция**

**Современные
информационные технологии
и ИТ-образование**

СБОРНИК ТРУДОВ

**Под редакцией
проф. В.А. Сухомлина**

**Москва
2009**

УДК [004:377/378] (063)
ББК 74.5я431 (0)+74.6я431 (0)
С56

Печатается по решению редакционно-издательского отдела факультета Вычислительной математики и кибернетики Московского государственного университета

Рецензенты: профессор, д.ф.-м.н. А. Н. Томилин

профессор, д.ф.-м.н. Л. А. Калиниченко

С56 Современные информационные технологии и ИТ-образование.
Сборник докладов научно-практической конференции: учебно-методическое пособие. Под ред. проф. В. А. Сухомлина.
– М.: ИНТУИТ.РУ, 2009. – 848 с.

ISBN 978-5-9556-0115-1

В сборник трудов включены доклады IV Международной научно-практической конференции «Современные информационные технологии и ИТ-образование», прошедшей в Московском государственном университете им. М. В. Ломоносова 14-15 декабря 2009 г. Целью конференции являлась интеграция усилий университетов, науки, индустрии и бизнеса в решении актуальной задачи построения востребованной национальной системы ИТ-образования.

Материалы сборника предназначены для научных работников, преподавателей, аспирантов и студентов, интересующихся проблемами ИТ-образования, теоретическими, методологическими и прикладными вопросами в области информационных технологий.

УДК [004:377/378](063)
ББК 74.5я431(0)+74.6я431(0)

ISBN 978-5-9556-0115-1

Система событийного программирования SEvents

Р.М.Биков, студент, УдГУ

Введение

Согласно одному из определений программирование - это искусство. И есть множество книг, обучающих этому искусству с разных точек зрения. Однако, на практике существует множество стилей программирования и порой, для того, чтобы полностью использовать возможности выбранного стиля, необходимо несколько изменить способ своего мышления. В случае, если человек долго работает в одном стиле, становится виртуозным программистом, ему сложнее потом переучиваться для работы в другом. Неискущённому же человеку изучение различных стилей даётся проще. В последствии человек, знающий различные стили программирования, в конкретной ситуации сможет выбрать наиболее подходящий стиль и оптимальнее решить стоящую перед ним задачу. В связи с этим возникает проблема обучения студентов (возможно и школьников) различным стилям программирования.

Из всего разнообразия стилей программирования условно можно выделить следующие:

- автоматный;
- структурный;
- функциональный;
- сентенциальный;
- событийный.

Событийный стиль характеризуется тем, что в нём вычисления активизируются в результате какого-либо события в системе, являются реакцией на это событие, изменяя характеристики (состояние) системы [1]. Событийный стиль может быть 2-х вариантов: программирование от событий и программирование от приоритетов.

Для первых 4-х стилей существуют языки и системы, их поддерживающие, которые могут быть использованы для обучения студентов соответствующему мышлению.

Событийное же программирование присутствует практически в каждой программе, однако поддерживается в виде специальных вставок пакетами, которые малопригодны для обучения, так как они перегружены множеством деталей, не относящихся к сути дела.

В связи с этим возникла задача создания системы событийного программирования, которая могла бы быть использована для обучения студентов событийному программированию. При проектировании системы

главными условиями были полное журналирование изменений, простота логики, концептуальная непротиворечивость.

SEvents¹ - система событийного программирования с фактором случайности, в большей степени является реализацией программирования от приоритетов, эта система кроссплатформена, с открытым исходным кодом, создана специально для обучения. По сути своей она близка к системе имитационного моделирования.

Описание системы

В связи с тем, что система изначально проектировалась для учебных целей, при реализации необходимо было максимально очистить событийное программирование от ненужных деталей и упростить логику взаимодействия событий. Кроме того, необходимо было придумать язык программирования, упрощающий описание событий. С целью обеспечения кроссплатформенности системы для её реализации был выбран язык Java.

Программа на языке SEvents имеет вид сценария. Впервые такой подход к описанию событий был применён в играх компании Paradox Entertainment, где использовался открытый сценарий.

Заголовок сценария SEvents представляет собой описание переменных и констант 3-х типов: boolean (логический тип), int (целочисленный), double (вещественный). В системе невозможно объявлять массивы и другие структуры. После заголовка идёт описание событий.

Каждое событие представляется в виде заголовка и тела события.

Заголовок события состоит из описания следующих параметров:
имени события;
условия выполнения события;
приоритета события;
вероятности выполнения события;
флага возобновимости события.

Первые два параметра являются обязательными, так как имя события является однозначным идентификатором события, а условие выполнения – это триггер, включающий активацию события и его обработку в системе.

Тело события состоит из последовательности операторов, которые должны выполняться по одному в единицу времени в случае активации события. Фактически в теле события описывается его обработчик. В теле события отсутствуют циклы и ветвления.

Операторы в системе могут быть двух типов:

¹ Систему можно найти по адресу <http://sourceforge.net/projects/sevents>

простейший оператор вычисления (вычисление какого-либо выражения, изменение значений переменных);
оператор работы с событиями.

Основными операторами являются операторы работы с событиями, поэтому в обработчике события невозможно произвести какие-либо нетривиальные вычисления. Для нетривиальных вычислений к системе можно подключать дополнительные модули – динамические библиотеки, которые одновременно являются объектами системы, так как имеют возможность реагировать на состояние системы в каждый момент времени, также могут изменять состояние системы. Объектам предоставлена возможность визуализации в окне программы.

В системе реализованы следующие операторы работы с событиями:
`int getDelay(“событие”) – возвращает оставшееся время задержки выполнения события;`
`boolean isFinished(“событие”) – возвращает true, если событие закончилось, false иначе;`
`boolean isRun(“событие”) – возвращает true, если событие выполняется, false иначе;`
`boolean isSleep(“событие”) – возвращает true, если событие деактивировано, false иначе;`
`void killEvent(“событие”) – безвозвратно деактивирует событие;`
`void setDelay(“событие”, int delay) – деактивирует событие на delay шагов;`
`void setSleep(“событие”) – деактивирует событие на неопределённое количество шагов;`
`void setWake(“событие”) – активирует событие, деактивированное на неопределённое количество шагов;`
`boolean wasEvent(“событие”) – возвращает true, если событие успело завершиться хотя бы один раз, false иначе.`

Объекты системы также имеют возможность воспользоваться операторами событий. Взаимодействие объектов и системы происходит посредством Java Native Interface. В силу технических особенностей реализации перед выполнением сценария для каждой библиотеки должен генерироваться свой класс, содержащий имя вызываемого при передаче управления объекту метода библиотеки.

Пример описания события:

```
event
    name = "Event";
    cond = (x > y) && (wasEvent("init e") || time > 30);
    probability = 0.5;
    priority = 20;
    repeatable = true;
```

```
body
  x = (y * 2 - 5) / z;
  b = b || wasEvent("First");
  setSleep("Second");
  setWake("Third");
end
```

Для моделирования взаимодействия событий используется внутренняя переменная времени *time*, которая пробегает от 0 до максимального значения. В каждый момент времени проверяются условия выполнения всех событий, в случае истинности условия выполнения с заданной вероятностью событие активируется и ставится в очередь на выполнение, из очереди на выполнение события извлекаются согласно их приоритетам и в каждом событии выполняется один очередной оператор в один момент времени. События с равным приоритетом выполняются в произвольном порядке в единицу времени. Однако каждое событие с некоторой вероятностью может выполняться наряду с событиями более высокого приоритета.

Притом при изменении значений переменных в сценарии или объектами мгновенное обновление значений происходит с некоторой вероятностью. Гарантированное обновление значений переменных происходит перед изменением переменной времени. Таким образом, моделируются нежелательные эффекты параллелизма.

Заключение

Система событийного программирования создана для того, чтобы научить студентов думать в рамках событий. Эта система является обучающей с альтернативной обучающей парадигмой, она даёт возможность делать много ошибок, анализировать и исправлять их, в результате выработать свой стиль. Данная программа на данный момент используется в учебном процессе УдГУ.

Литература

1. Непейвода Н.Н., Стили и методы программирования, Интернет-Университет Информационных Технологий, Москва, 2005.