

УДК 004.415.2

А.В. Лезарев, А.С. Шкуркин

ОБЪЕКТНАЯ МОДЕЛЬ ПОДСИСТЕМЫ МАТЕМАТИЧЕСКОГО МОДЕЛИРОВАНИЯ¹

В рамках разработки «Объектно-ориентированной распределенной системы моделирования процессов массового обслуживания» рассматривается задача проектирования подсистемы математических расчетов. Предложена концептуальная схема модуля. Для объектного представления формул построена диаграмма классов. Рассмотрена последовательность действий при вычислении значения выражения.

Ключевые слова: объектно-ориентированное проектирование, математическое моделирование.

При разработке системы «Объектно-ориентированная распределенная система моделирования процессов массового обслуживания» [1] было выдвинуто требование производить независимо от имитационного и математическое моделирование систем массового обслуживания и представлять сравнительные результаты на примере конкретных систем. Для выполнения расчетов, не нуждающихся в имитационном моделировании, расчетов, имеющих аналитические или численные решения, предлагается использовать компонент, отвечающий за алгоритмы математического моделирования (АММ).

1. Общая структура модуля

В процессе разработки была получена концептуальная схема модуля, общий вид которой представлен на рис. 1.

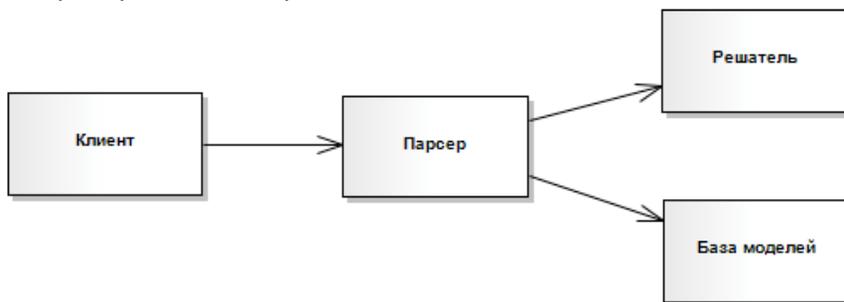


Рис. 1. Структура модуля «Алгоритмы математического моделирования»

Подсистема математических расчетов содержит три компонента. Компонент *Парсер*, являющийся связующим звеном между *Базой моделей*, *Решателем* и клиентами, предназначен для получения команд от клиентов на проведение вычислений и возвращения назад полученных результатов.

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

Для выполнения расчетов создается соответствующее представление математических формул, записываемое в объектном виде, что позволяет упростить работу по конструированию формул, сохранение формулы и ее восстановление из базы моделей, а также формирование команды для решателя. Объектное представление формул используется также в подсистеме *Конструктор* системы [1] для создания выражений, с помощью которых рассчитываются определенные характеристики систем массового обслуживания.

Компонент *Решатель* предназначен непосредственно для математических вычислений. Предполагается реализация собственных механизмов вычисления, а также обращение в случае необходимости к внешним компонентам – библиотекам алгоритмов или математическим пакетам (например, MatLab). Для поддержки возможности смены решателя в компоненте *Парсер* будут реализованы механизмы для перевода объектного представления формулы в команду конкретного решателя. При проектировании данной подсистемы использовано типовое архитектурное решение «Шлюз» [2].

Для хранения описания моделей и их характеристик совместно с выражениями, по которым они вычисляются, предназначен компонент *База моделей*, реализация которого планируется в виде XML-файла.

2. Схема математических расчетов

Рассмотрим общую схему функционирования подсистемы математических расчетов. *Парсер* получает от *Клиента* команду на расчет определенной характеристики. В качестве параметров передаются название модели (идентификатор), название характеристики и значения параметров модели. После этого *Парсер* находит в базе моделей соответствующую запись и получает объектное представление формулы. Далее формируется команда для решателя. *Решатель* производит соответствующие расчеты и возвращает значение. Затем *Парсер* передает решение *Клиенту*.

Рассмотрим более подробно один из главных модулей компонента *Парсер*, отвечающий за создание объектного представления формулы. Задачи, возникающие при расчете числовых значений характеристик систем массового обслуживания, разделим на два класса: вычисление значения выражения и численное решение уравнений. На данном этапе рассматривается только первая задача.

Для представления значений параметров и результатов создадим специальный тип данных *DataType*. Он будет использоваться как для записи чисел, так и матриц. К тому же элементами матрицы могут быть не только числа, но и переменные, и математические выражения. Диаграмма классов иерархии типа *DataType* приведена на рис. 2.

Для объектного представления математической формулы применяем паттерн объектно-ориентированного проектирования «Компоновщик» [3]. На рис. 3 представлена полученная диаграмма классов объектной модели математической формулы.

Класс *FormulaAtom* инкапсулирует общий интерфейс всех объектов математических расчетов. Операция *Calculate()* предназначена для получения окончательного значения выражения. Реализация этой операции возлагается на потомков – класс *Variable* и иерархию классов *Expression*. Последние, в свою очередь, содержат в себе более мелкие структурные блоки выражения, также являющиеся потомками *FormulaAtom*.

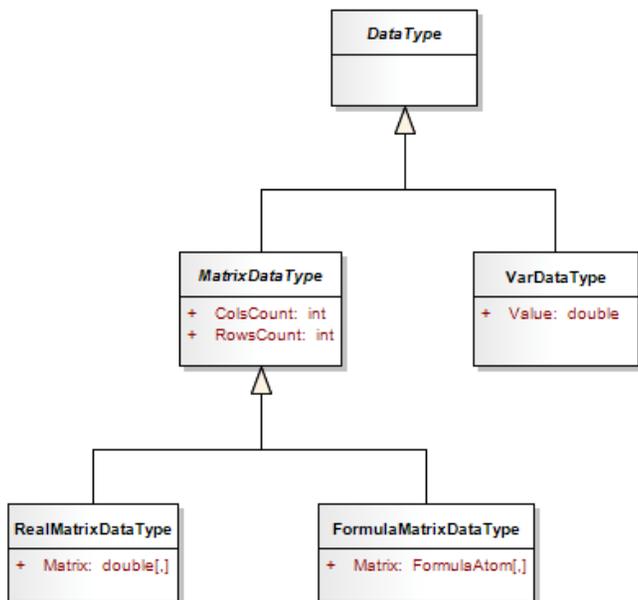
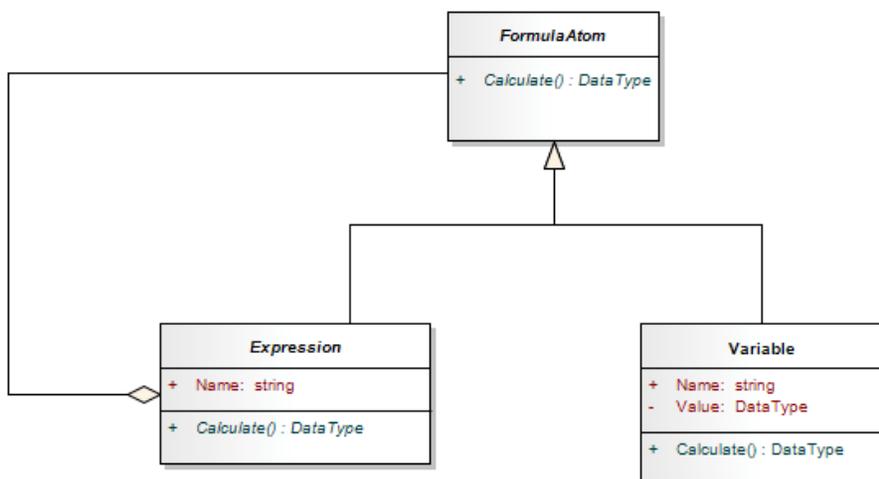
Рис. 2. Диаграмма классов типа *DataType*

Рис. 3. Диаграмма классов объектной модели математической формулы

Класс *Variable* предназначен для представления переменных и констант, входящих в запись формулы. В поле *Value* хранится значение переменной или константы. Поле *Name* предназначено для хранения символического обозначения переменной. Данный класс используется как для описания скалярных значений, так и для описания матриц. Метод *Calculate()* возвращает значение переменной.

Класс *Expression* определяет операции и функции, входящие в запись формулы. В поле *Name* хранится символическое обозначение выражения. От класса *Expression* наследуются классы для вычисления конкретной функции или опера-

ции. В этих классах определяется необходимое количество аргументов и конкретизируется метод *Calculate()*. Примеры наследников класса *Expression* приведены на диаграмме (рис. 4).

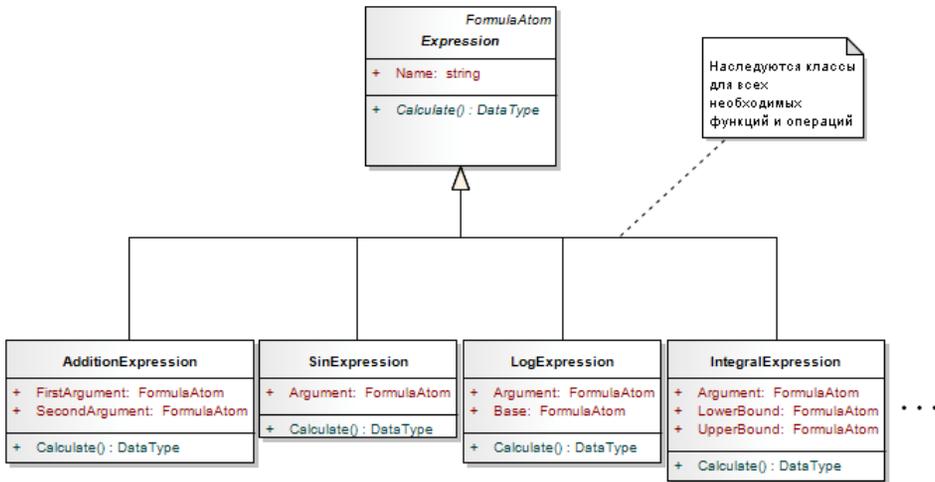


Рис. 4. Структура наследования класса *Expression*

Каждый наследник определяет прямые ссылки на переменные и выражения, входящие в его состав (являющиеся его аргументами или параметрами). Например, для вычисления логарифма (класс *LogExpression*) требуются два аргумента: основание задается с помощью поля *Base*, а подлогарифмическое выражение – *Argument*.

Для понимания процесса вычисления формул рассмотрим его на примере вычисления логарифма. На рис. 5. приведена диаграмма последовательности для процесса вычисления формулы.

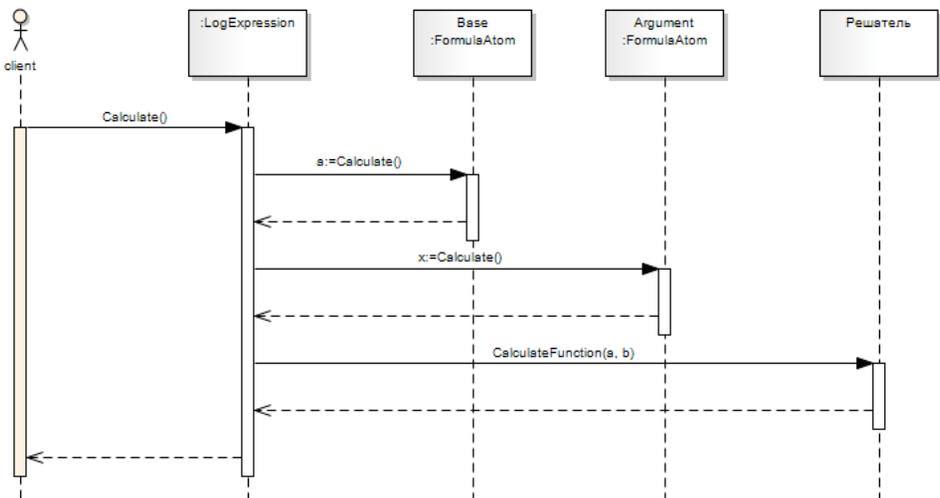


Рис. 5. Диаграмма последовательности «Вычисление формулы»

Процесс взаимодействия начинается с того, что внешний субъект *Client* (в нашем случае это может быть и объект подсистемы, например, стоящий выше по иерархии объект формулы) обращается к операции *Calculate()* экземпляра *LogExpression*. Выполнение самой операции *Calculate()* начинается с последовательного вычисления аргументов. Для этого вызываются методы *Calculate()* объектов *Base* и *Argument*. Последние, в свою очередь, проделывают аналогичные операции над своими внутренними подвыражениями. Такой рекурсивный вызов продолжается до тех пор, пока метод *Calculate()* не будет вызван у экземпляра класса *Variable*, который возвратит конкретное числовое значение (или числовую матрицу). После того как будут готовы все промежуточные значения, вызывается соответствующая функция *Решателя*, которая производит непосредственные расчеты и возвращает результат. Затем результат передается объекту *Client*.

Заключение

Таким образом, в работе представлена концептуальная модель подсистемы математических расчетов, а также рассмотрены вопросы объектного представления математических выражений. Предложена структура модуля для работы с объектным представлением формул, рассмотрена последовательность действий, выполняемых при вычислении значений выражений.

ЛИТЕРАТУРА

1. *Войтиков К.Ю., Моисеев А.Н.* Модель компонентов объектной распределенной системы моделирования процессов массового обслуживания // Информационные технологии и математическое моделирование (ИТММ-2009): Материалы VIII Всероссийской научно-практической конференции с международным участием (13-14 ноября 2009 г.). Томск: Изд-во Том. ун-та, 2009. С. 122 – 124.
2. *Фаулер М.* Архитектура корпоративных программных приложений: пер. с англ. М.: Издательский дом «Вильямс», 2004. 544 с.
3. *Гамма Э., Хелм Р., Джонсон Р., Влиссидес Дж.* Приемы объектно-ориентированного проектирования. Паттерны проектирования. СПб: Питер, 2007. 366 с.

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

Лезарев Александр Викторович

Шкуркин Алексей Сергеевич

Анжеро-Судженский филиал Кемеровского государственного университета

E-mail: lezarev@asf.ru; shkurkin@asf.ru

Поступила в редакцию 21 января 2010 г.