ОБЪЕКТНО-ОРИЕНТИРОВАННЫЙ ПОДХОД В ПАРАЛЕЛЬНОМ МОДЕЛИРОВАНИИ

Н. Ю. Чеботарев
Кафедра ЭВМ, ДонГТУ
E-maiI. vader@cs.dgtu.donetsk.ua

Abstract

Chebotaryov N. Object-oriented method in parallel simulation. Article is overview of object-oriented method of physical system simulation. There are reviews of object orientation role in simulation, questions of implementation of object-oriented simulation tools and support model life cycle with metaclasses.

Введение

Известны три подхода к моделированию непрерывных и непрерывно-дискретных динамических систем с использованием современных вычислительных средств [1]:

- блочно-ориентированный (БО);
- уравнение-ориентированный (УО);
- объектно-ориентированный (00).

БО- и УО-подходы реализованы в виде языков моделирования SIMULrNK [2], ISRSIM [3], ACSL [4], которые доступны широкому кругу пользователей персональных компьютеров и рабочих станций. Общим методическим свойством БО- и УО-подходов является то, что они требуют преобразования системы уравнений моделируемых объектов к виду, удобному для моделирования: дифференциальные уравнения разрешаются относительно старших производных, а алгебраические - относительно искомых переменных. БО-модель строится в виде блок-схемы соединением операционных блоков, необходимых для решения системы уравнений объекта. УО-модель записывается на входном языке в виде последовательности алгоритмических операторов, решающих систему уравнений объекта. Состав блоков и операторов БО- и УО-языков моделирования принципиально одинаков. Их программная реализация также аналогична: как функциональные блоки, так и операторы имплементируются в виде подпрограмм на языках С и Fortran, а формирование моделей из подпрограмм осуществляется интерпретаторами.

ОО-моделирование динамических систем находится в стадии развития [1, 5, 6]. Рассмотрим вопросы объектной ориентации построения моделей динамических систем, которые важны для разработки параллельных распределенных моделирующих сред.

1. Роль объектной ориентации в моделировании

Актуальность перехода в моделировании к ОО-средствам обусловлена следующими факторами. Во-первых, интенсивное развитие объектно-ориентированного проектирования привело к необходимости перестроить все САПР ОО-способом. При этом было замечено, что в составе ОО-САПР должны быть подсистемы моделирования, являющиеся наиболее эффективным средством проверки проектных решений. Интеграция этих средств с ОО-САПР возможна лишь на общей ОО-основе. Во-вторых, моделированию как интердисциплинарной науке объектная ориентация позволит преодолеть ряд методических сложностей, которые имеют место В БО- и УО-языках моделирования и которые ярко себя проявили в процессе разработки параллельных моделирующих сред [7, 8].

Так, ОО-метод позволяет строить наглядные модели как совокупности объектов, взаимодействующих через свои интерфейсы посредством передачи сообщений друг

другу. Объекты могут абстрактно отражать физические сущности компонентов моделируемой системы, а могут также реализовывать математические операции, необходимые для решения уравнений. Таким образом, существует возможность строить модели как на основе их математического описания, так и на основе объектно-технологического описания структуры и поведения компонентов моделируемой системы. Алгоритм поведения объектов может быть легко изменен с помощью механизмов наследования и полиморфизма. Особенности физических процессов и математическое описание объектов могут быть инкапсулированы внутри базовых классов библиотек предметных областей. ОО-метод предлагает простую структурную организацию модельной программы, которая сохраняет смысловые связи между абстракциями предметной области, абстракциями языка и других средств описания модели. Такая организация позволяет сохранить соответствие между визуальным и языковым описаниями моделируемых систем. За основу языков последовательного и параллельного моделирования динамических систем может быть взят объектно-ориентированный язык программирования, поддерживающий, помимо основных, ряд дополнительных концепций ОО-парадигмы: параллелизм, устойчивость и типизацию.

Построение ОО-языка моделирования, имплементируемого в SISD-, вШО- и MIMD-архитектурах на единой методической основе, существенно повысит уровень дружественности к пользователям современных средств моделирования динамических систем.

2. Реализация средств ОО-моделирования

Средства ОО-моделирования являются частью подсистемы диалога массивно параллельной моделирующей среды (МПМС) [7, 8, 9]. На основе структуры подсистемы диалога МЛМС [9] может быть предложена следующая структура ОО-системы моделирования (рис. 1).

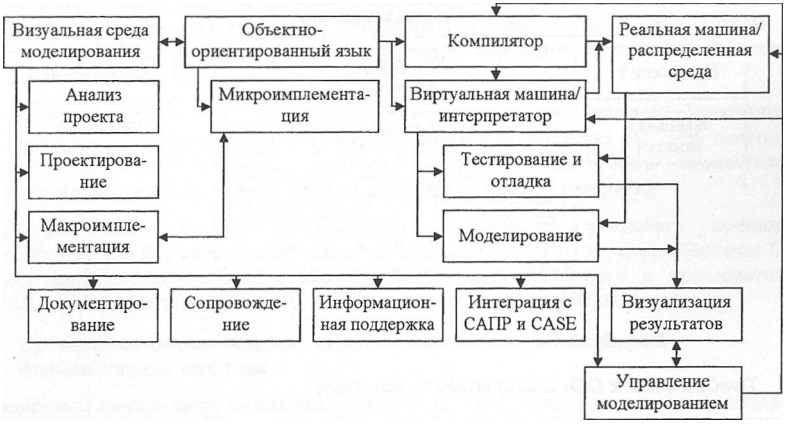

Рисунок 1. Структура объектно-ориентированной системы моделирования

В данной структуре центром управления является визуальная среда моделирования. Она состоит из средств управления жизненным циклом модели (средств анализа, проектирования, макроимплементации, документирования и сопровождения), средств информационной поддержки (интерактивной помощи, руководств пользователя, подсистемы обучения и т.д.), средств интеграции с другими средами разработки (САПР

и CASE) и средств управления моделированием и визуализации результатов. Основным средством описания модели является объектно-ориентированный язык, на который отображается макроимплементация и с помощью которого выполняется микроимплементация, т.е. описываются детали внутренней структуры и поведения объектов.

Тестирование, отладка и моделирование могут выполняться на различных аппаратно-системных платформах, в том числе, в распределенной вычислительной среде [1, 8]. Возможна реализация, при которой на реальной машине могут работать как виртуальная машина (ВМ) или интерпретатор, так и откомпилированная программа. В распределенной вычислительной среде модель может работать только на виртуальной машине или интерпретаторе.

Виртуальная машина может соответствовать MIMD-архитектуре [10]. На остальных же системах и в распределенной среде естественная МЕШЭ-параллельность будет эмулироваться виртуальной машиной. Можно предложить следующую укрупненную архитектуру виртуальной машины (рис. 2). Как видно из рисунка, ВМ представляет собой сочетание МШГО-системы с общей и локальной памятью [10], где каждый процессор реализует конвейерную архитектуру и может производить вычисления над массивными данными, реализуя SEVTO-архитектуру [10]. Реализация суперскалярных процессоров может быть сделана на физическом уровне какой либо вычислительной платформы. Естественно предполагается, что язык ОО-моделирования будет поддерживать данную архитектуру ВМ посредством реализации механизма синхронизации потоков (например, подобно языку Java [11]) и механизма массовых параллельных вычислений (например, подобно языку Parallaxis [10]).

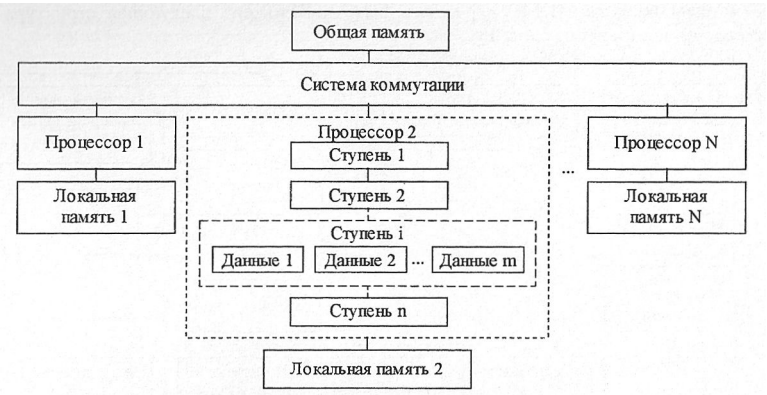

Рисунок 2. Укрупненная архитектура виртуальной машины

3. Требования к ОО-языку моделирования

Можно предъявить следующие основные требования к 00-языку моделирования. Объектно-ориентированный метод основан на четырех основных (абстракция, инкапсуляция, модульность, иерархия) и трех дополнительных концепциях (типизация, параллелизм, устойчивость) [6], Поддержка основных концепций обязательна, поскольку без них модель не будет объектно-ориентированной. Поддержка типизации рекомендуема для обеспечения простого и защищенного от ошибок управления приведением типов, а также для эффективной реализации полиморфизма. Поддержка параллелизма необходима для эффективной реализации процесса моделирования на параллельных системах. При этом поддержки простого параллелизма,

ориентированного на управление потоками, как это сделано в Java, будет явно недостаточно. Существует необходимость организовывать массовые параллельные вычисления (матрично-векторные операции) в рамках одного потока управления на системах, поддерживающих вШЛ-архитектуру. Поддержка устойчивости необходима по следующим причинам:

- может существовать необходимость выполнять моделирование в несколько этапов с сохранением промежуточных состояний модели;
- моделирование производится в вычислительной среде, где обмен данными осуществляется посредством устойчивых объектов.

В объектно-ориентированной модели могут существовать шесть типов иерархии: ассоциация, наследование, агрегация, использование, инстанцирование, метаклассы [6].

Поддержка ассоциаций (смысловых связей между абстракциями), т.е. отношений типа «один-к-одному», «один-ко-многим», «многие-ко-многим», необходима для организации целостных моделей. Также можег существовать необходимость хранения каких-либо наборов пассивных данных, организованных реляционным способом.

Наследование должно быть множественным для создания возможности многократного использования имеющихся интерфейсов и/или реализаций. Множественное наследование необходимо и для организации многоаспектного представления модели (например, математического, семантического, проблемно-ориентированного).

Поддержка агрегации необходима для построения составных объектов. Более того, без агрегации нельзя говорить об инкапсуляции.

Отношения использования (например, клиент-сервер) необходимо поддерживать, т. к. к этой форме иерархии преобразуются некоторые ассоциации. Без отношений использования невозможно построить реальные модели.

Поддержка аппарата инстанцирования (например, механизма шаблонов, подобно языку C++ [6, 12]) рекомендуема для сокращения объема кода и для устранения необходимости в макроопределениях, которые нарушают стройность объектно-ориентированной модели.

Метаклассы необходимы для обеспечения манипуляций классами (описаниями классов) как объектами во время выполнения (моделирования) [б]. Это позволяет строить мутагенные модели, изменение которых может быть осуществлено пользователем или подсистемой самой модели по заданному алгоритму.

При реализации средств ОО-моделирования следует учитывать принципы спецификации и формализации функций сложных систем [13], которые обеспечат ОО-языку моделирования более высокий уровень дружественности к пользователям (экспертам предметных областей) и преемственность с БО- и УО-языками.

4. Средства сопровождения жизненного цикла ОО-моделей динамических систем

Жизненный цикл модели динамической системы (МДС) предлагается характеризовать периодами, содержащими этапы разработки и применения.

Период П-1 характерен теоретической работой в предметной области ДС и содержит этапы:

- концепция;
- физическое (технологическое) описание элементов ДС;
- математическое описание физических (технологических) процессов в элементах ДС;
- описание топологии (структуры ДС);

- математическое описание физических (технологических) процессов во всей структуре ДС с учетом законов взаимодействия элементов ДС.

Период П-2 должен быть посвящен экспериментальным исследованиям ДС самой предметной области с этапами:

- создание экспериментальной базы (установки, физические модели, прототипы технологических процессов, лабораторные анализы и др.) или подготовка к экспериментам на реально работающих объектах;
- разработка планов экспериментов, направленных на идентификацию и оценку достоверности моделей ДС;
- проведение экспериментов и регистрация динамических процессов.

Период П-3 - реализация моделей ДС по их формальному описанию (топологическому и математическому) с этапами, подробно изложенными ***в*** методических материалах средств моделирования всех поколений - от приведения описаний к требуемому виду до решения уравнений и визуализации результатов. Разрабатываемые моделирующие среды относятся к новейшим средствам обеспечения этого периода существования моделей сложных ДС [1].

Период П-4 - идентификация моделей ДС путем сравнения результатов экспериментов (П-2) и моделирования (П-3), возможная корректировка формального описания модели.

Период П-5 - это активное использование модели для решения задач проектирования, реализации и внедрения динамических систем, а также задач исследовательского характера по совершенствованию теории и созданию ДС нового поколения.

Параллельная моделирующая среда должна иметь дружественные к пользователю средства, активно поддерживающие работу с моделями ДС во всех периодах их жизненного цикла. Эта работа может иметь оперативный интерактивный характер (тестирование, визуализация значений, изменение параметров, изменение топологии и состава уравнений, использование готовой модели в составе различных проектов и др.) или эпизодический с общим интервалом возможной активизации, который может составлять несколько лет (изменение концепций, переход на ДС нового поколения, переход на новую вычислительную базу, приход новых сотрудников и др.). Основной проблемой здесь может оказаться иерархичность модели с множеством подсистем и ее изменяемость во времени (мутагенность).

Для решения этой проблемы может быть использован механизм метаклассов. Метаклассы позволяют манипулировать целостными формальными описаниями моделей, элементами интерфейсов и реализаций классов как объектами. Работа с моделью выполняется в следующем порядке:

- при наличии метакласса для нужного класса, поля, метода и т.д. получается соответствующее описание;
- производится изменение этого описания;
- останавливаются активные объекты изменяемых классов;
- производится фиксация изменений: это может быть сделано вызовом специальной функции реконструктора, которая задает алгоритм преобразования из одной формы в другую;
- запускаются ранее активные объекты.

Для интерактивного вмешательства пользователя в модель во время ее выполнения средства разработки/моделирования должны обеспечивать соответствующий интерфейс доступа (примером такой среды разработки может служить ГОМ VisualAge for Smalltalk [14, 15]).

Реконструкция объектов в принципе может быть выполнена и без остановки активных объектов, но в обоих случаях могут возникать сложности синхронизации между активными и останавливаемыми/реконструируемыми потоками.

Заключение

Объектно-ориентированный подход к разработке параллельных моделирующих сред позволит преодолеть ряд методических сложностей, которые имеют место В существующих языках моделирования. Рассмотрены роль объектной ориентации в моделировании, вопросы реализации средств ОО-моделирования и поддержки жизненного цикла моделей динамических систем в моделирующей среде с помощью метаклассов. Следующим этапом работы в этом направлении является ОО-имплементация компонент моделирующей среды.

Литература

1. Святний В. А. Проблеми паралельного моделювання складних динамічних систем, Всб. «Новые технологии - путь в будущее», АТН Украины, Донецк, 1999, 104-113
2. MATLAB - SIMULD4K: eine Leistungsfähige Umgebung fflr die Simulation nichtlinearer dynamischer Systeme. The MathWorks inc. 1996.
3. ISRSIM: Simulationssystem fur dynamische Systeme mit graphischer Bedienobertläche. Version 5.0, Universität Stuttgart, 1990.
4. ACSL for Windows User's Guide, Revision 11.4.4, April 1997.Copyright © 1997 by MGA Software.
5. Schmidt, B: Simulationssysteme der 5. Generation. SiP, Heftl, 1994, 5-6
6. Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений на С++, 2<. изд. / Пер. с англ. - М.: «Издательство Бином», СПб.: «Невский диалект», 1999 г. - 560 с., ил.
7. Anoprienko A., Svjatnyi V., Braunl T., Reuter A., Zeitz M., Massiv parallele Simulationsumgebung ffir dynamische Systeme mit konzentrierten und verteilten Parametem-Simulationstechnik. 9. Simposium in Stuttgart, Oktober 1994, Tagungsband, Vieweg (1994), 183-188.
8. Feldmann L., Svjateiyj V., Lapko V., Gilles E.-D., Zeitz M., Reuter A., Rothhermel K.: Parallele Simulationstechnik. Наукові праці ДонДТУ, Серія "Проблеми моделювання та автоматизованого проектування динамічних систем", вип. 10, Донецьк, ДонДТУ, 1999, с. 9-19
9. Чеботарев Н. Ю. К разработке подсистемы диалога MIMD-компоненты массивно параллельной моделирующей среды. / Научные труды Донецкого государственного технического университета. Серия: «Информатика, кибернетика и вычислительная техника», Выпуск 6 - Донецк: ДонГТУ, 1999, с. 62-66
10. Бройнль Томас. Паралельне програмування: Початковий курс: Навч. посібник / Вступ, слово А. Ройтера; Пер. з нім. В. А. Святного. - К.: Вища шк , 1997. - 358 с.: іл.
11. Java™ 2 SDK, Standard Edition. Documentation. Version 1.2.2. Copyright O 1995-1999 Sun Microsystems, Inc.
12. Страуструп Б. Язык программирования С++, 3-е изд. / Пер. с англ. - СПб.; М.: «Невский диалект» - «Издательство БИНОМ», 1999 г. - 991 с., ил.
13. Чеботарев Н. Ю. Проблемы спецификации в процессе разработки сложных информационных систем. / Наукові праці ДонДТУ. Серія: "Проблеми моделювання та автоматизованого проектування динамічних систем", вип. 10, Донецьк, ДонДТУ, 1999, с. 231-234
14. Shafer, Dan. IBM Smalltalk Programming For Windows and OS/2, California. Prima Publishing, 1995.
15. Smith, David N. IBM Smalltalk: The Language. New York. The Benjamin/Cummings Publishing Company, Inc. 1995.