

УДК 519.3

Объектно-ориентированный подход к параллельному моделированию химико-технологических процессов

Бондарева Е.С.

Донецкий национальный технический университет
kbondareva@gmail.com

Abstract

Bondareva E. Object-oriented approach to parallel modeling of chemical processes. There is considered the object-oriented approach to parallel modeling of chemical processes. The analysis and modeling tools requirements are described, features of user interface for parallel models' design, which correspond to the requirements of the parallel simulation environment, are listed.

Введение

Ограниченность возможностей аналитического исследования таких систем, как химико-технологические процессы, делает актуальным применение метода объектно-ориентированного моделирования, который позволяет представить описание этих процессов в соответствующей форме.

Процессы химической технологии строго классифицированы и имеют математическое описание, которое позволяет представлять их в виде типовых компонентов технологических процессов с заданной информацией о диапазонах изменений значений входных переменных, параметров технологического режима и т.д.

Подобная классификация предоставляет возможность исследовать сложные системы, концентрируя основное внимание на объектах систем. С этой позиции объектно-ориентированный подход (ОО-подход), как средство создания моделей химико-технологических процессов, является современным решением для методики их построения [1,2].

Подходы к построению параллельных моделей химико-технологических объектов и технологически-ориентированная параллельная моделирующая среда описаны в [3,4], где предложено представление технологического процесса в виде модульной иерархической структуры и рассмотрены связанные с этим задачи топологического анализа и генерирования уравнений.

В данной статье рассматривается задача объектно-ориентированного подхода к построению параллельных моделей химико-технологических процессов и ее имплементация в соответствии с требованиями, предъявляемыми к среде моделирования. Описывается создание библиотек основных блоков и моделей и возможности пользовательского интерфейса для составления параллельных моделей в режиме, дружественном к исследователю моделей.

Краткий анализ и требования к средствам моделирования

Существующие моделирующие среды разделяются на две большие группы (рис. 1).

К первой группе относятся проблемно-ориентированные моделирующие среды, специализирующиеся на решении специфических задач конкретной прикладной области (горного дела, металлургии, электротехники, химической технологии, биотехнологий и т.д.).

Ко второй относятся универсальные среды, ориентированные на определенный класс математических моделей и применимые для любой прикладной области, в которой эти модели пригодны.

Обе группы могут быть представлены моделирующими средами компонентного моделирования.

К универсальным средам также относятся математические пакеты моделирования. В математических пакетах (Mathematica, MathCAD, MatLab, Maple [5]) предполагается, что математическая модель всей моделируемой системы уже каким-либо образом построена и ее требуется исследовать. Такой подход характерен в основном для научных исследований. Как правило, математические пакеты сочетают численные эксперименты с символическими преобразованиями. Модель здесь исследуется как математический объект, а проблемы ее представления и формы записи не относятся к первостепенным по отношению к проблемам изучения ее свойств, параметров и поиска других методов решения.

Сложные динамические системы (СДС), в которых состав компонентов изменяется во времени, не могут быть решены в пакетах математического моделирования.

Таким образом, для СДС построение и сопровождение ее полной математической модели вручную практически невозможно. Компонентное моделирование предполагает, что описание

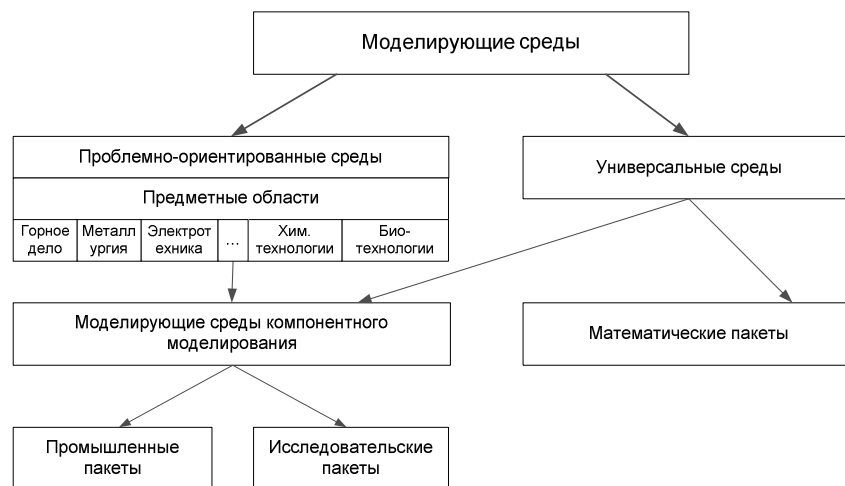


Рисунок 1 – Моделирующие среды

моделируемой системы строится из компонентов (в том числе и готовых библиотечных), а совокупная математическая модель формируется пакетом автоматически. Моделирующие среды компонентного моделирования в основном ориентированы на численные эксперименты и могут быть разделены по способам их применения или технологии моделирования на две группы.

Пакеты первой группы относятся к пакетам промышленным или к коммерческим, которые могут являться как универсальными, так и проблемно-ориентированными моделирующими средами. При использовании таких моделирующих сред ведущую роль играет организация работ: хорошо налаженное взаимодействие между отдельными группами, быстрый доступ к многочисленным экспериментальным данным и библиотекам программ, тщательное документирование и тестирование, многовариантные расчеты. При этом обычно используются хорошо изученные готовые математические модели, которые лишь модифицируются и приспособляются для решения конкретных задач.

Пользователи пакета подразделяются на две категории: разработчики библиотек готовых моделей и обычные пользователи, работа которых сводится к составлению схем из типовых блоков и параметрическая настройка блоков. К таким пакетам можно отнести, к примеру, Блочный ориентированный язык моделирования SIMULINK пакета MATLAB [6-8], который является наиболее известным современным представителем направления блочного моделирования, ориентированного на графический язык иерархических блок-схем. Модель здесь собирается из имеющихся в библиотеке функциональных блоков, обеспечивающих решение уравнений. Собранный схему

можно объявить типовым блоком следующего уровня – подсистемой.

Унифицированный объектно-ориентированный язык моделирования физических систем Modelica [9-11] позволяет пользователю самому задавать описание новых компонентов, определяя их поведение в виде системы дифференциально-алгебраических уравнений. Modelica – свободно распространяемый объектно-ориентированный язык для моделирования сложных физических систем, который объединил в себе результаты, достигнутые в области моделирования электрических, механических, гидравлических и других систем. Компиляторы языка являются коммерческими. Благодаря объектно-ориентированному подходу, созданные на языке Modelica модели легко модернизировать и создавать на их основе более сложные экземпляры. Спецификация языка не предусматривает поддержки создания параллельных моделей.

В коммерческих пакетах [12,13] также реализованы возможности оптимизации производства, анализа существующих проблем и аварийных ситуаций, оценка экономических аспектов производства, планирование ресурсов, продукции, графика работ и т.д. Однако подобное моделирующее программное обеспечение имеет очень высокую цену лицензии и на данный момент не позволяет проводить эксперименты с моделью с использованием параллельных архитектур.

Рассмотрим вторую группу моделирующих сред компонентного моделирования – группу исследовательских пакетов, предназначенных для предварительных исследований, выполняемых отдельными учеными или проектировщиками [14-16]. Библиотеки готовых моделей используются здесь ограниченно. Исходным материалом служат плохо формализованные

моделі, чьи свойства еще не вполне изучены.

Исследовательские пакеты уступают по количеству уникальных возможностей промышленным. Но они более просты для освоения и доступны отдельному исследователю при решении задач, посильных одному или нескольким разработчикам, не являющимся специалистами в области программирования и вычислений. С исследовательскими пакетами тесно связана концепция активного вычислительного эксперимента, позволяющего максимально быстро оценивать вариант исследуемой модели [17], поэтому здесь вопросы формы представления модели, ее построения, играют важную роль.

Средства моделирования технологических процессов должны предоставлять необходимый инструментарий для проведения каждого этапа моделирования, а способы использования каждого из этих инструментов должны носить дружественный к разработчику модели и пользователю характер, что является одним из основных требований к разработке моделирующих сред. Разработчику модели должны быть обеспечены средства общения с системой в формах, быстро им распознаваемых, т.е. на языке, знакомом пользователю. Например, текст, условная графика, образная графика (объекты в виде графических примитивов) и т.д. Это требование приводит к объектно-ориентированному подходу в разработке средств построения параллельных моделей технологических процессов.

Объектно-ориентированный подход

При объектно-ориентированном моделировании (ОО-моделирование) система рассматривается как группа объектов, которые в процессе взаимодействия друг с другом определяют поведение системы в целом. Понятие объекта является здесь центральным. Объект абстрактно отображает физическую сущность компонент модели. Так, существует возможность построения объектной модели химико-технологического процесса на основе описания структуры и поведения компонент исследуемого объекта.

Построение объектной модели основано на следующих принципах:

- абстрагирование,
- инкапсуляция,
- модульность,
- иерархия,
- типизация,
- параллелизм,
- устойчивость.

Поддержка основных концепций

обязательна, так как без них модель процесса не будет соответствовать объектно-ориентированной.

Обнаружение общих абстракций и механизмов технологии химического производства позволяет создавать библиотеки различных типовых элементов химической технологии, как для стандартных задач проведения эксперимента, так и для новых задач исследования. Это в значительной степени облегчает моделирование сложной системы за счет использования разработанных ранее элементов, а также уменьшает затраты времени на создание новых моделей с другими свойствами и параметрами.

Преимуществами ОО-моделирования химико-технологических процессов являются [18,19]:

1. Создание библиотек моделей, что позволяет решать ряд методических задач моделирования, а именно: создавать библиотеки типовых компонентов как библиотеки классов; повторно использовать компоненты с помощью наследования классов; строить модели с множеством однотипных объектов; выполнять параметризацию моделей с помощью полиморфизма; создавать и уничтожать экземпляры объектов в ходе вычислений.

2. Снижение затрат на построение моделей исследуемых объектов за счет независимой разработки и тестирования. Каждая часть ХТП может моделироваться и исследоваться изолированно. Реализация каждого модуля не должна зависеть от реализации других модулей. Это приводит к меньшей сложности и эффективной тестируемости применяемых модулей.

3. Свободный выбор численных методов в модулях. Для каждого модуля в отдельности может применяться именно тот численный метод, который в состоянии решить уравнения модуля с соблюдением заданных требований.

4. Интеграция с САПР. Другой важной мотивацией использования ОО-подхода является все более широкое использование объектно-ориентированного анализа при разработке сложных технических систем. В настоящее время происходит тенденция к процессу интегрирования систем моделирования с ОО-САПР, так как система моделирования в данном случае является наиболее эффективным средством проверки готового продукта процесса проектирования. Интеграция системы моделирования с ОО-САПР возможна лишь на общей ОО-основе.

5. Применение параллельных вычислительных архитектур. Данный подход сокращает затраты времени на проведение эксперимента с моделью. Применение параллельного моделирования обеспечивается как аппаратными, так и программными средствами моделирующей среды.

Архитектура пользовательского интерфейса

В технологически-ориентированной параллельной моделирующей среде реализованы следующие возможности для составления параллельных моделей:

- создание элементарных и составных блоков;
- использование графической формы описания модели;
- интерактивная поддержка выбора решателя;
- интерактивная поддержка анализа распараллеливания модели;
- визуализация результатов параллельного моделирования.

Основным элементом графического представления параллельной модели является технологическая схема процесса, построенная из образов отдельных компонентов, соединенных функциональными связями.

Среда предоставляет пользователю модели возможность построения классификации элементарных блоков – типовых компонентов, из которых создаются составные блоки или общая модель, а также возможность конструирования из блоков имеющихся типов структурированной иерархичной модели. При этом каждый блок, который может представлять определенную технологическую установку, является экземпляром своего класса. Класс блока задает обобщенный прототип, который является типовым компонентом. Экземпляр блока определяет некоторое конкретное устройство, соответствующее заданному классу. Таким образом, иерархия классов позволяет выделить обобщенные свойства элементов данной модели в виде типовых компонентов, собранных в библиотеки.

Типовые компоненты, представляющие различные по функциональному назначению классы, делятся на группы классов и собраны в библиотеку групп классов. В графической оболочке они визуальнo отображаются как дерево типовых компонентов. В результате выполнения операции drag-and-drop (переместить графический объект в заданное место при помощи мыши) графический образ класса переносится из библиотеки классов в визуальное хранилище блоков - графический редактор. При этой операции происходит создание экземпляра класса со значениями параметров, заданными по умолчанию, которые при необходимости могут быть изменены.

При создании блока вся информация о нем сохраняется в файл в формате XML. Визуальный образ блока задается пользователем на этапе его создания. Значения параметров блока задаются при его создании и могут быть изменены при

создании экземпляров данного блока, т.е. параметры доступны извне. Пользователь может изменять значения параметров в окне параметров каждого блока (рис.2).

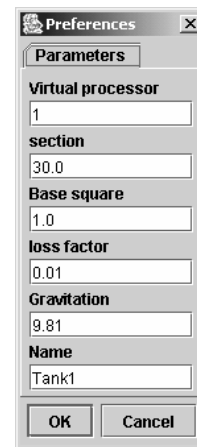


Рисунок 2 – Окно параметров

В общем случае при создании элементарного блока в его описании содержатся следующие элементы: параметры, информация о входах и выходах, переменные, уравнения, внутренняя структура и графическое описание.

Вкладка редактора уравнений при создании элементарного блока выглядит как показано на рисунке 3.

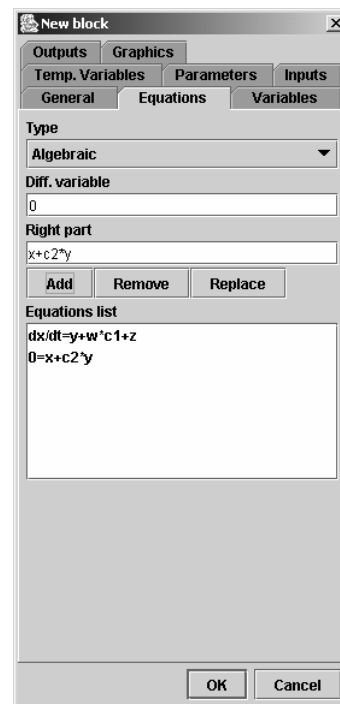


Рисунок 3 – Редактор уравнений блока

Составные блоки строятся в графическом редакторе из экземпляров библиотечных классов, соединенных функциональными связями, образующими функциональную схему составного

блока. В таком составном блоке внутренние блоки могут взаимодействовать между собой, а также с входами/выходами внешних блоков посредством компонентов, задающих точки входа/выхода для составных блоков.

Интерфейс составного блока также как и элементарного представлен входами, выходами и параметрами, которые при его создании задаются переопределением входов, выходов и параметров входящих в него блоков. Переопределение входов и выходов задается во время построения функциональной схемы составного блока. Параметры переопределяются во время создания описания составного блока. Таким образом, часть локальных переменных составного блока инкапсулирована в его теле, а другая часть переменных объявляется интерфейсными, что делает их доступными для других блоков.

Совокупное поведение составного блока определяется совместным функционированием включенных в него внутренних блоков с учетом функциональных связей. Если при создании составного блока внутренние блоки также являются составными, то имеет место иерархическая функциональная схема, что соответствует технологическим схемам процесса.

Основным элементом графического представления исследуемой модели является структурная схема, построенная из образов отдельных компонентов, соединенных функциональными связями. Исследуемая пользователем модель технологического процесса может быть представлена как четырехуровневая иерархическая система, информация о которой должна быть сохранена в модели [3]. Исходя из вышесказанного, модель вообще может быть представлена верхним уровнем иерархии – в виде одного блока. Описание модели – блоков, входящих в нее, и функциональных связей – сохраняется в xml-файл. Данное описание хранит ссылки на входящие в модель блоки, где задано описание тела блока, интерфейсные переменные – параметры для отображения в окне редактирования параметров – и список функциональных связей модели, используемых в формировании топологической информации, требуемой для построения систем уравнений модели [4].

В технологически-ориентированной параллельной моделирующей среде использован принцип открытых интерфейсов, позволяющих подключать различные библиотеки численных методов, расположенных на удаленных серверах, для решения полученных генератором уравнений систем уравнений.

На рисунке 4 показана диаграмма взаимодействия стандарта UML клиентской части и сервера, предоставляющего интерфейс, который реализует методы доступа к данным и управления процессом моделирования.

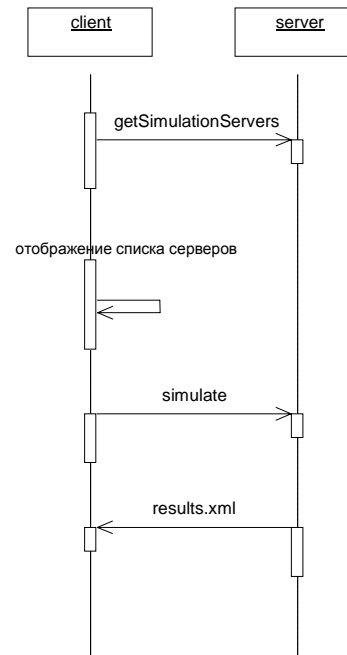


Рисунок 4 – Диаграмма взаимодействия клиента и сервера

Тип решателя и сервер, на котором он расположен, а так же его адрес и порт задаются в файле формата XML. Данный файл размещается на сервере, предоставляющем интерфейс метода *getSimulationServers*, к которому обращается клиентское приложение для получения списка доступных на данный момент серверов. Пользователь перед началом процесса моделирования имеет возможность выбирать решатель, а также задавать следующие параметры моделирования: требуемый шаг для моделирования и время процесса моделирования.

Выбранные пользователем сервер для моделирования и параметры моделирования передаются серверу с помощью метода *Simulate*, реализованного в интерфейсе сервера, после чего он производит запуск на моделирование. По окончании процесса моделирования данные с результатами моделирования в формате XML передаются клиенту для визуализации.

Пользователь имеет средства задания блоку его виртуального процессора, на котором должно происходить моделирование. Реализована возможность редактирования данного параметра в окне задания параметров блока (рис.2)

Пользователю также предоставлены средства для визуализации результатов моделирования. Таким средством является использование блока «score», который выполняет функцию осциллографа, т.е. пользователь имеет возможность просмотра графика изменения во времени моделирования параметров выхода, к которому подсоединен осциллограф. По

окончании процесса моделирования данные с результатами моделирования, а именно данные значений в момент времени выходов, к которым был подсоединен блок «score», в формате XML передаются сервером клиенту для визуализации.

интерфейса для параллельного моделирования химико-технологических процессов, который отличается удобством составления параллельных моделей, и соответствует требованиям, предъявляемым к среде параллельного моделирования.

Заключення

Проведен анализ средств моделирования и описаны возможности пользовательского

Литература

1. D. Anagnostopoulos An object-oriented modelling approach for dynamic computer network simulation / D. Anagnostopoulos, M. Nikolaidou // International Journal of Modelling and Simulation – 2001. – Volume 21, Issue 4 – P.249 - 256.
2. Буч Г. Объектно-ориентированный анализ и проектирование с примерами на C++ / Буч Г. – [3-е изд. / пер. с англ.] – М.: «Издательство Бином», СПб.: «Невский диалект», 2001. – 560 с.
3. Святный В.А., Бондарева Е.С. Топологічний аналізатор технологічно-орієнтованого паралельного моделюючого середовища.// Наукові праці ДонНТУ, серія МАП. Випуск: 78 – Донецьк: ДонНТУ. – 2005, с.5-12.
4. Бондарева Е.С. Генератор уравнений параллельной моделирующей среды для технологических процессов / Бондарева Е.С. // Наукові праці ДонНТУ, серія «Інформатика, кібернетика та обчислювальна техніка» – 2008. – Вип. 9 – С. 93-98.
5. Чарльз Генри Эдвардс. Дифференциальные уравнения и проблема собственных значений: моделирование и вычисление с помощью Mathematica, Maple и MATLAB / Чарльз Генри Эдвардс, Дэвид Э. Пенни – [3-е изд.] – М.: «Вильямс», 2007. – С. 203-219.
6. Simulink: Dynamic System Simulation for MATLAB – Using Simulink / MathWorks Inc. – [Руководство пользователя] – 1999. – [3rd edition].
7. Черных И.В. Simulink: среда создания инженерных приложений. / Черных И.В. – М.: ДИАЛОГ-МИФИ, 2003. – 496 с.
8. Дьяконов В. Simulink 4 / Дьяконов В. – С. Петербург: Питер, 2002. – 528 с.
9. Elmqvist H. Modelica – the new object-oriented modeling language / Elmqvist H., Mattsson S.E., Otter M. // The 12th European Simulation Multiconference – 1998. – P.127-131.
10. Fritzson P. MathModelica – an extensible modeling and simulation environment with integrated graphics and literate programming / Fritzson P., Gunnarson J., Jirstrand M. // 2nd International Modelica Conference – 2002. – P. 41-54.
11. Modelica - A Unified Object-Oriented Language for Physical Systems Modeling – Version 2.0 – [Электронный ресурс] – Режим доступа : <http://www.modelica.org/documents/ModelicaSpec20.pdf>
12. Официальный сайт компании Aspen Technology, Inc., Cambridge. [Электронный ресурс] – Режим доступа: <http://www.aspentech.com>
13. ASPEN PLUS User Guide. Version 10 – [Руководство пользователя] – 1998. – Vol.1.
14. DIVA-3.9 User Manual / [M. Häfele, A.Kienle, E.Klein, A.Kremling, C.Majer, M.Mangold, A.Spieker, E.Stein, R. Waschler, K.P. Zeyer] – 2002
15. Krasnyk, M. The ProMoT/Diana simulation environment / Krasnyk, M.; Bondareva, K.; Milokhov, O.; Teplinskiy, K.; Ginkel, M.; Kienle, A. // 16th European Symposium on Computer-Aided Process Engineering and 9th International Symposium on Process System Engineering – 2006. – Volume 21A – P. 445-450.
16. Святный В.А. Паралельне моделювання складних динамічних систем / Святный В.А. // Наукові праці ДонНТУ, серія “Проблеми моделювання та автоматизації проектування динамічних систем” – 2006. – вип.5 – С. 5-20.
17. Колесов Ю.Б. Программная поддержка активного вычислительного эксперимента / Колесов Ю.Б., Сениченков Ю.Б. // Научно-технические ведомости СПбГПУ. – 2004. – №1.
18. Marquardt W. Modellbildung und Simulation verfahrenstechnischer Prozesse / Marquardt W. – Lehrstuhl für Prozeßtechnik – Rheinisch-Westfälische Technische Hochschule Aachen – 1996.

Поступила в редакцию 29.03.2010