

Генератор уравнений параллельной моделирующей среды для технологических процессов

Е.С. Бондарева
 Донецкий национальный технический университет
 Кафедра ЭВМ ДонНТУ
 kbondareva@gmail.com

Abstract

Bondareva E. Equation generator of parallel simulation environment for engineering process modeling and simulation. There is considered the approach for process technology model parallelization and equation generator implementation in this article. This approach is available for solution of complexity problem of process technology. The main purpose of this article is to consider the levels of parallelization of the model, their analysis and equation generator implementation.

Введение

Сложный тесно связанный технологический процесс описывается системой, состоящей из нескольких тысяч дифференциально-алгебраических уравнений. Проблема моделирования технологических процессов сводится к решению этих больших систем уравнений, затратных по времени вычисления. Решением данной проблемы является создание параллельной модели исследуемого технологического процесса для последующего моделирования на параллельных архитектурах с использованием специализированных численных методов. Применение данного подхода позволяет снизить временные затраты на моделирование [1, 2, 3].

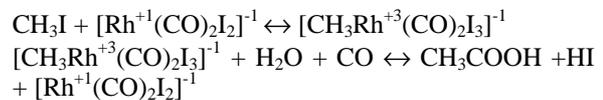
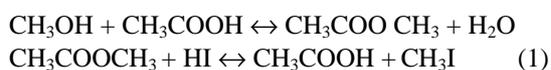
В данной статье рассматривается подход к построению параллельной модели технологического процесса на примере реактора синтеза уксусной кислоты и связанный с этим процесс генерирования системы уравнений.

Краткое описание технологической схемы реактора синтеза уксусной кислоты

Уксусная кислота является продуктом реакции метанола и окиси углерода, проходящей в газожидкостном реакторе с механическим перемешиванием с использованием катализатора и активатора в гомогенной жидкой фазе при температуре 185° С и давлении 28 атм [4,5].

Схематическое изображение газожидкостного реактора синтеза уксусной кислоты представлено на рисунке 1.

Механизм получения уксусной кислоты состоит в реализации четырех промежуточных реакций (1):



, где CH_3COOH – уксусная кислота,
 H_2O – вода,
 CH_3I – йодистый метил,
 HI – йодистый водород,
 $\text{CH}_3\text{COOCH}_3$ – метилацетат,
 Rh – родий,
 $[\text{Rh}^+(\text{CO})_2\text{I}_2]^{-1}$ – карбонильный комплекс родия

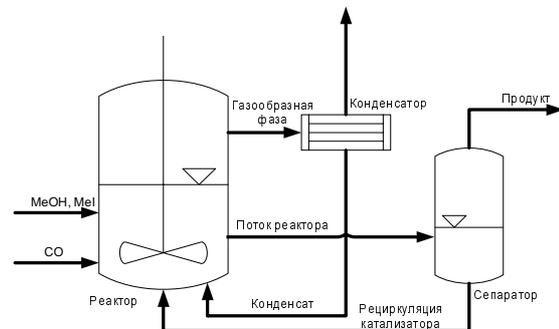


Рисунок 1 – Схематическое изображение реактора

Химические процессы, происходящие в реакторе, характеризуются высокой степенью сложности и многообразием влияющих на ход процесса параметров, а также сложной кинетикой химических реакций и нелинейностью характеристик. Математическая модель всех основных реакций с учетом их динамики является нелинейной и представлена в виде системы, состоящей из 24 дифференциально-алгебраических уравнений [6].

Этапы построения параллельной модели технологического процесса

Методику построения параллельной модели технологического процесса предлагается

вести в соответствии с основными этапами, последовательность которых приведена на рисунке 2.

Исходными данными здесь являются результаты топологического анализа, которые представлены в работе [7], где предложено поэтапное разбиение технологического процесса, имеющего модульную иерархическую структуру, на более мелкие составляющие блоки или модули, рассматриваемые как самостоятельные элементы.

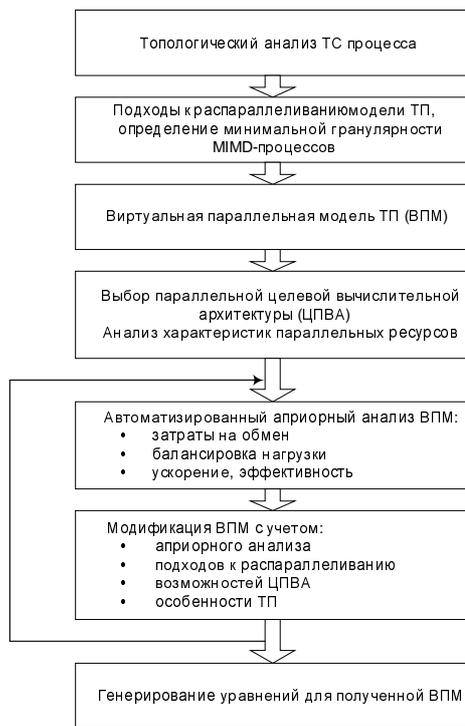


Рисунок 2 – Этапы разработки параллельной модели

Модульная структура модели технологического процесса задается пользователем с помощью объектно-ориентированного подхода [8]. Данный подход позволяет применить такие возможности как абстракция, наследование и полиморфизм для описания отдельных модулей. Задание связей между модулями позволяет составить иерархическую модель исследуемого технологического процесса.

Результатом такого модульного представления модели является автоматическое получение топологической информации – информации о связях между блоками технологического процесса, представленной в виде общей матрицы связей ТС. Данное автоматическое разбиение осуществляется топологическим анализатором, являющимся частью программного обеспечения параллельной моделирующей среды [9]. Информация, находящаяся в общей матрице связей ТС, кроме связей между блоками, позволяет удобно определить:

- внутренние и внешние связи для блока любого уровня декомпозиции (аппарат, фаза, уравнение)

- уровень иерархии и, соответственно, уровень вложенности блока

Информация о декомпозиции, а также информация о типах связей блока и его вложенности тесно связана с понятием о минимальной гранулярности процессов. Поэтому дальнейшими шагами являются выбор подхода к распараллеливанию модели технологического процесса, определение минимальной гранулярности виртуального процесса и получение виртуальной параллельной модели (ВПМ). Рассматриваемые далее этапы относятся к процессу девиртуализации, которые содержат следующие основные функции: выбор параллельной целевой архитектуры, анализ ее характеристик; автоматизированный априорный анализ виртуальной параллельной модели в соответствии с критериями балансировки загрузки, затрат на передачу данных, ускорением, эффективностью. Важным этапом является модификация ВПМ с учетом всех результатов априорного анализа, а также возможных подходов к распараллеливанию и количества процессоров в целевой вычислительной архитектуре. Здесь также должны быть учтены особенности математической модели исследуемого технологического процесса. Модификация ВПМ является процессом итеративным, таким образом, должна существовать возможность возврата на более ранний этап – этап априорного анализа модели. Генерирование уравнений для ВПМ осуществляется генератором уравнений.

Минимальная гранулярность процессов. Девиртуализация и уровни распараллеливания

Объем вычислений, который выполняется в заданном процессе, характеризует гранулярность виртуального процесса. Параллельная модель технологического процесса может быть представлена на различных уровнях иерархии в зависимости от того, как будет задана гранулярность процессов в модели. Данный принцип иллюстрируется на рисунке 3 на примере реактора синтеза уксусной кислоты.

Исходя из структуры исследуемого объекта технологического процесса, уравнений, описывающих его компоненты и их элементы, рассмотрим следующие возможные уровни грануляции процессов и MIMD-подход к распараллеливанию, основанный на модульной структуре модели технологического процесса.

- 1 процесс = 1 уравнение

Третий, самый нижний уровень иерархии, представляет простейшие неделимые далее блоки, описываемые уравнениями таких

термодинамических явлений как процесс сохранения баланса вещества и энергии. На данном уровне может быть задана минимальная гранулярность процесса, где одному процессу ставится в соответствие решение одного

уравнения. На данном уровне следует учитывать, что количество связей между процессами максимально, и, следовательно, на обмен данными может быть затрачено времени больше, чем на само решение поставленной задачи.

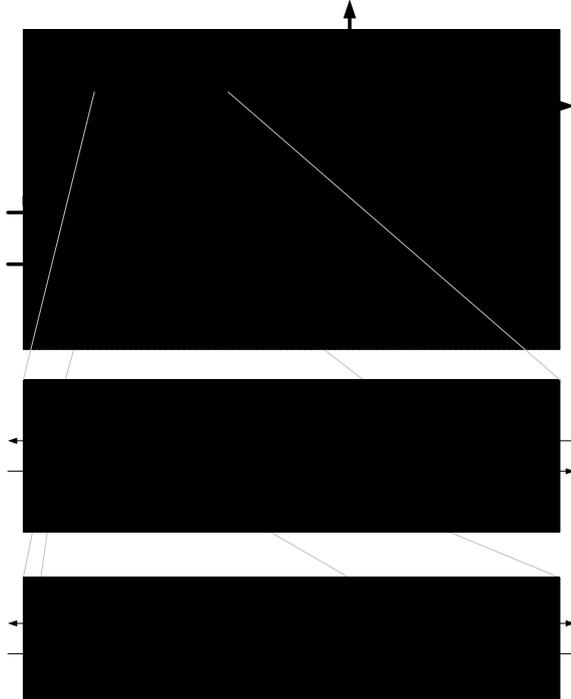


Рисунок 3 – Гранулярность процессов

- 1 процесс = 1 фаза ТП

Предлагаемый второй уровень иерархии представлен уровнем термодинамической фазы. Так как реактор является газожидкостным, то данный уровень для него будет представлен двумя блоками, описывающими жидкую и газообразную фазы. Этот уровень является системой из компонент третьей, самой нижней ступени иерархии, т.е. используется принцип объектно-ориентированной композиции. Здесь один процесс включает систему уравнений, описывающих газообразную и жидкостную фазу. Следовательно, на данном уровне иерархии параллельная модель является менее детализированной, и один процесс может содержать целую процедуру или подпрограмму вычисления системы уравнений, что напрямую соответствует принципу MIMD-параллельности. Из общей матрицы связей, являющейся результатом топологического анализа объекта, возможно получить информацию об уравнениях, относящихся к определенной фазе и наиболее связанных между собой. Таким образом, один подобный блок содержит внутри себя наиболее тесно связанные компоненты, а количество внешних связей является для него минимальным.

- 1 процесс = 1 аппарат ТС

Первый уровень, являющийся самым верхним уровнем иерархии, представлен уровнем аппарата ТС. Соответственно, здесь один процесс

включает такие аппараты ТС как реактор, сепаратор и конденсатор, которые на данном уровне представляются как элементарные компоненты. Данные компоненты описываются сложными математическими моделями, представленными композицией компонентов соседнего нижнего уровня, т.е. взаимодействием термодинамических фаз аппарата ТС. Рассматриваемый уровень является крупнозернистым уровнем распараллеливания, так как представлен множеством операций, и также соответствует принципу MIMD-параллельности.

Следует отметить, что при данном подходе к распараллеливанию разбиение модели объекта целесообразно проводить на стадии топологического анализа, с учетом критериев априорного анализа. Это позволяет снизить количество обменов между компонентами, учитывая показатели балансирования загрузки и минимизации затрат на передачу данных.

Для автоматизированного анализа достаточно информации из топологического анализатора, формирующего общую матрицу связей, но, когда количество уравнений достигает больших значений, манипулирование отдельными уравнениями становится ресурсоемким и сортировка общей системы занимает значительное время. Поэтому предполагается, что уравнения, находящиеся внутри одного блока,

связаны между собой гораздо сильнее, чем уравнения в разных блоках. Таким образом, на данном этапе существует возможность манипулировать целыми блоками, а не уравнениями, что значительно облегчает задачу. Учитывая то, что модель представляется иерархической системой с несколькими уровнями, то существует возможность регулировать минимальную зернистость в модели, как показано ранее, вплоть до отдельных уравнений. Для реализации данного подхода необходима информация о принадлежности тех или иных уравнений конкретным блокам от топологического анализатора, формирование которой предусмотрено при построении топологической информации модели.

Разделение системы на подсистемы на требуемом уровне распараллеливания, после того

как гранулярность процесса определена, может осуществляться путем минимизации общих переменных между блоками и максимально равномерного распределения уравнений по блокам, а также с учетом количества вычислительных узлов параллельной целевой вычислительной архитектуры.

Основной целью алгоритма является минимизация количества всех внешних соединений между блоками, пока не будут получены блоки, максимально равные по размеру. Таким образом, результатом сортировки строк и столбцов в матрице связей (рис.4) является ее упорядоченный блочно-диагональный вид с максимально одинаковыми по размеру блоками и минимальным количеством внешних, относительно этих блоков, связей.

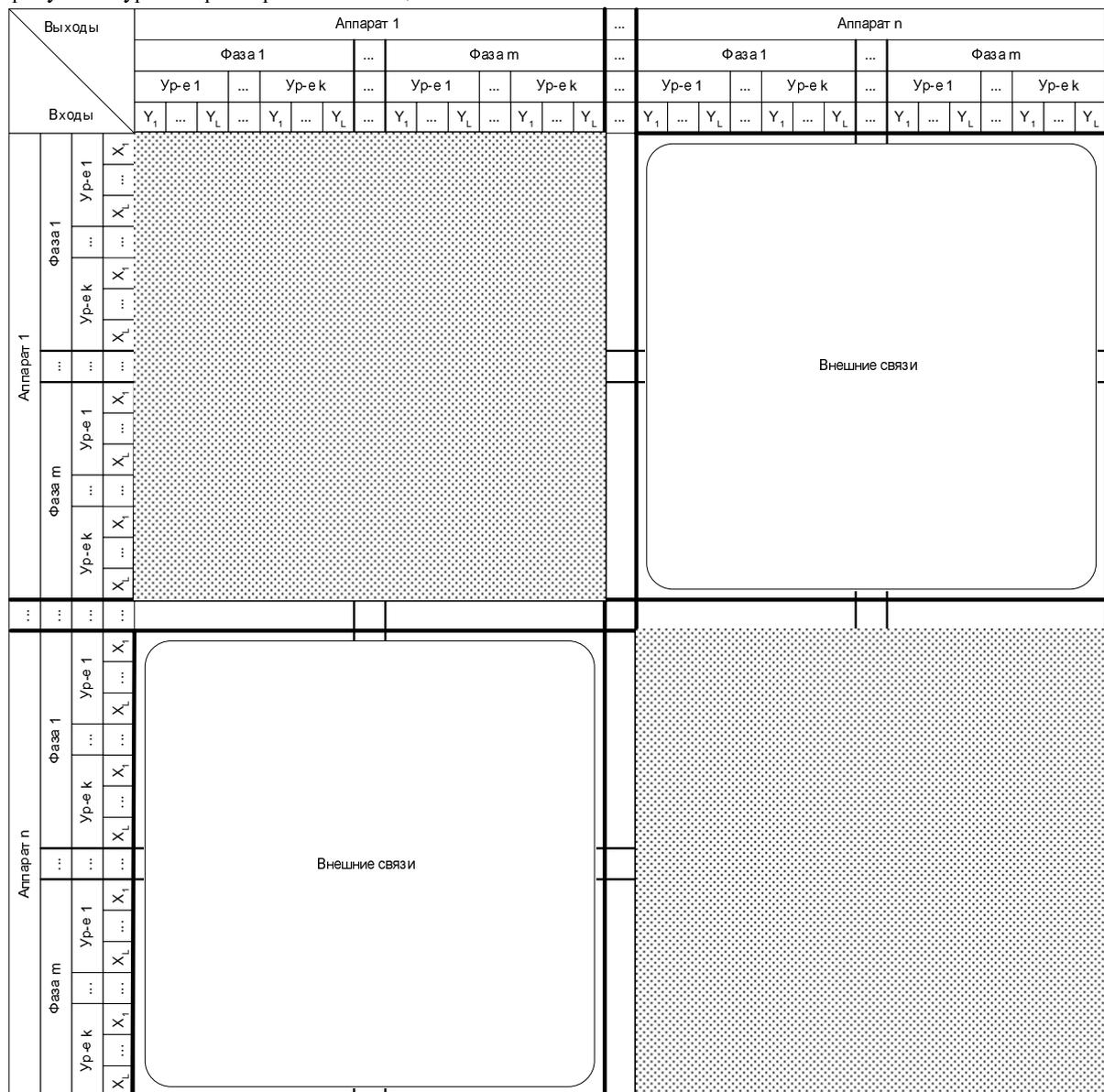


Рисунок 4 – Блочно-диагональный вид матрицы связей

Таким образом, результатом деvirtualизации ВПМ являются блоки, максимально равные по размерам и минимально связанные между собой.

Генератор уравнений

Одним из основных вопросов данной работы является имплементация автоматического преобразования заданных пользователем уравнений модели к виду, который позволяет проводить численное интегрирование моделируемого объекта.

Форма записи уравнений модели, доступная пользователю, обычно отличается от формы, необходимой численному алгоритму, и определяется существующим программным обеспечением [10, 11]. Каждая заданная пользователем система уравнений моделируемого объекта, сопровождаемая набором вспомогательных уравнений, должна быть приведена к общему виду, необходимому для программной реализации численного метода, а формулы отсортированы в правильном порядке.

Таким образом, на следующем этапе решается проблема автоматического получения математической системы согласно разбиению модели и топологической информации - описания связей. Связи несут информацию о соответствии выходных переменных одного блока входным переменным другого (рис.4). Наличие связи между ними означает, что входное значение X равно выходному значению Y . Вход X является приемником в функциональной связи и может участвовать только в одной функциональной связи. В этом случае значение входа-приемника всегда равно значению источника в этой связи. Выход Y может являться источником нескольких функциональных связей. Функциональные связи являются естественной абстракцией для описания соединений в технологических схемах, и топологическая информация о них используется в связывающих уравнениях для генерирования совокупной системы уравнений созданной модели.

Предлагаемый генератор уравнений является программным модулем автоматического получения виртуальной параллельной модели, преобразованной к виду, удобному для применения численных методов (Simulation model) (Рис.5).

После получения общей системы уравнений и топологической информации, уравнения, заданные на входном языке, транслируются в форму, пригодную для компиляции на языке программирования (C++), а соответствующая система уравнений структурируется в подсистемы уравнений согласно полученному разбиению. Далее генерируются подпрограммы обмена данными для компонентов виртуальной параллельной

модели, которые создаются на основе информации, полученной от топологического анализатора; задаются идентификаторы для переменных таким образом, чтобы названия переменных были уникальными в общей системе уравнений. Также при генерировании создается информация о принадлежности уравнений блокам. Далее повторяющиеся вычисления константных выражений и неиспользуемые переменные могут быть удалены из кода генерирования уравнений, что позволяет провести предварительную обработку системы уравнений без использования решателя уравнений. Оптимизация на данном уровне приводит к повышению продуктивности моделирования и более эффективному численному решению.

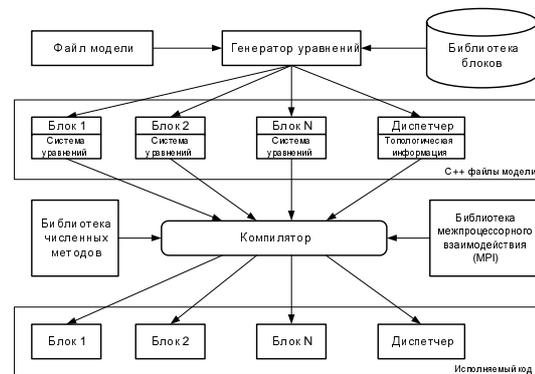


Рисунок 5 – Структура генерирования уравнений

Результатом генерирования является исполняемый код каждой из подсистем параллельной модели, полученный путем компиляции сгенерированной подсистемы на языке высокого уровня (C++) и подготовленный к параллельному решению численными методами. Особенностью является генерирование исполняемого кода согласно MPMD (Multiple Program Multiple Data) – принципу, который обеспечивает выполнение различных программ над разными потоками данных, относится к функциональному распараллеливанию и соответствует рассмотренному подходу к построению параллельных моделей. Здесь каждый вычислительный узел выполняет решение своей подсистемы уравнений, каждая из которых находится в отдельном исполняемом файле и решается отдельным решателем. Вследствие этого возникает проблема синхронизации и координации выполнения этих подсистем. Для решения этой проблемы в структуру генератора уравнений вводится понятие диспетчера. Диспетчер представляет собой отдельный процесс, который управляет всем процессом моделирования, а именно: инициализирует все процессы, решающие отдельные подсистемы, выполняет передачу по запросу входных переменных для подсистем, осуществляет

хранение и запись результатов моделирования после каждого шага интегрирования для выходных переменных. Взаимодействие диспетчера с процессом, представляющим блок *i*, представлено в виде UML sequence-диаграммы на рисунке 6.

После выполнения инициализации, перед каждым шагом моделирования, блок запрашивает у диспетчера значения входных переменных. Диспетчер, используя топологическую

информацию, анализирует доступность этой переменной в данный момент времени (блок, выход которого связан с данным входом, завершил очередной шаг интегрирования) и передает ее блоку *i*. Так как системы уравнений блоков решаются отдельными решателями – шаги интегрирования в них могут не совпадать – в таком случае диспетчер производит при необходимости интерполяцию запрашиваемого значения переменной в момент времени *t*.

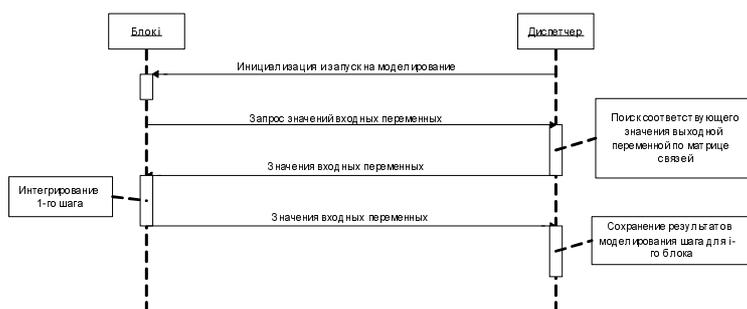


Рисунок 6 - UML sequence-диаграмма взаимодействия диспетчера с блоком

Следующим этапом в моделировании объекта технологического процесса является численное решение составленной параллельной модели, которое является перспективой дальнейших исследований в данном направлении.

Заключение

Подход к созданию параллельной модели объекта и генерации системы уравнений, предложенный в данной статье, позволяет решить проблему моделирования технологических процессов реальной сложности на параллельных архитектурах. Перспективой дальнейших исследований и разработок в данном направлении является использование решателей с разработкой алгоритмов управления численным решением параллельных моделей технологических объектов.

Литература

1. Святный В.А. Паралельне моделювання складних динамічних систем. Наукові праці ДонНТУ, серія "Проблеми моделювання та автоматизації проектування динамічних систем", вип. 5, 2006. – С. 5–20.
2. Аноприенко А.Я., Святный В.А. Высокопроизводительные информационно-моделирующие среды для исследования, разработки и сопровождения сложных динамических систем. Наукові праці ДонНТУ, серія "Проблеми моделювання та автоматизації проектування динамічних систем", вип. 29, 2001. – С. 346–267.
3. J.Borchardt. Newton-type decomposition

4. Acetic Acid and its Derivatives. Edited by Agreda V.H., Zoeller J.R. Marcel Dekker Inc., N.Y., 1993, 451 p.
5. Paulik F. E., Roth J. F. Chem. Commun., 1968, p. 1578.
6. Roland Waschler. Dissertation zur Erlangung des akademischen Grades Dr.-Ing., Magdeburg, 2005
7. 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.
8. Helget, A.: Modulare Simulation verfahrenstechnischer Anlagen. Dissertation. Fortschr.-Ber. VDI Reihe 20 Nr. 251. Düsseldorf: VDI Verlag 1997, 132S.
9. К.Бондарева, В.Святный. Verfahrenstechnisch orientierte parallele Simulationsumgebung.// Наукові праці ДонНТУ, серія МАП. Випуск: 5 (116) – Донецьк: ДонНТУ. – 2006, с.28-35.
10. Krasnyk, M.; Bondareva, K.; Milokhov, O.; Teplinskiy, K.; Ginkel, M.; Kienle, A.; The ProMoT/Diana simulation environment. In: European Symposium on Computer Aided Process Engineering <16, 2006, Garmisch-Partenkirchen>; 16th European Symposium on Computer-Aided Process Engineering and 9th International Symposium on Process System Engineering. - Amsterdam [u.a.]: Elsevier, S. 445-450; Computer-aided chemical engineering; 21 A
11. Modelica – Unified Object-Oriented Language for Physical Systems Modeling. Tutorial and Rationale. Version 1.4, December 15, 2000.