

ПРОГРАММНЫЙ КОМПЛЕКС ДЛЯ ПРОГРАММИРОВАНИЯ ИМИТАЦИОННЫХ МОДЕЛЕЙ ПРОИЗВОДСТВЕННЫХ СИСТЕМ

Р.В. Петров, С.А. Альховик

Аннотация. В статье представлен программный комплекс и методика создания имитационных моделей сложных производственных систем на основе процессного способа имитации с применением параллельных вычислений.

Ключевые слова: программный комплекс, имитационная модель, производственная система, процессный способ моделирования, параллельные вычисления.

1. ВВЕДЕНИЕ

Средства создания имитационных моделей (ИМ) являются востребованным в настоящее время, однако представленные на рынке системы моделирования зачастую не используют современные разработки в области информационных технологий. Существует класс сложных систем (СС), к которым относятся, например, производственно-экономические системы, создание ИМ которых является сложной и трудоемкой задачей как на этапах проектирования и реализации, так и при дальнейшем расширении и уточнении ИМ. Для решения указанной задачи разработаны и представлены на рынке различные программные комплексы (ПК), однако детальный анализ этих систем показывает, что для них характерны следующие недостатки [1, 2]:

- узкая специализация ПК, допускающая построение ИМ только определенного типа систем, другие типы систем либо не реализуются вовсе, либо их реализация требует значительных временных затрат;
- отсутствие модульности при разработке ИМ и невозможность многоуровневого описания, приводит к созданию монолитной модели, а следовательно увеличивается сложность изменения структуры модели и повторного использования отдельных ее частей;
- отсутствие поддержки ООП в языке моделирования, что приводит к усложнению построения ИМ;
- отсутствие поддержки этапа проектирования, что приводит к увеличению времени затрачиваемого на разработку ИМ в целом, так как разработчику приходится увязывать различные методологии при переходе к реализации ИМ;
- закрытость ПК, что не позволяет отдельно использовать ИМ в других разрабатываемых приложениях, а также создавать собственные надстройки над ПК;
- отсутствие поддержки распределенных вычислений.

Поэтому для эффективного построения ИМ производственной системы необходим инструмент, который бы устранял существующие недостатки, а также реализовывал методику создания ИМ, позволяющую не только описывать систему в терминах предметной области, но и переносить это описание на этап реализации. Исследование дискретных и непрерывных производственных процессов [3,4] показало, что наиболее оптимальным методом описания и реализации ИМ производственных систем является процессный метод. На основании этого спроектирован и реализован ПК и соответствующая методика разработки ИМ.

2. СТРУКТУРА ПРОГРАММНОГО КОМПЛЕКСА

Для создания ИМ наиболее оптимальным является использование существующей среды разработки. ИМ разрабатывается на основе базовой библиотеки классов и результатом является исходный текст и скомпилированный файл ИМ. Это позволило сократить время разработки ПК, так как не требуется разрабатывать свой язык программирования, компилятор или интерпретатор, а также позволяет повысить скорость работы ИМ. Использование среды разработки позволяет выполнить требование об использовании существующего объектно-ориентированного языка для описания модели.

Современные языки программирования поддерживают такие концепции как ООП, шаблоны, метаданные, делегаты, что позволяет существенно упростить процесс программирования.

В качестве основной платформы разработки ПК используется платформа .NET, которая имеет в своем составе все необходимые средства для реализации управляющей программы моделирования (УПМ), позволяет существенно упростить процесс разработки ИМ. Например, используя возможность динамически анализировать структуру программы во время ее выполнения, можно исключить дополнительное, помимо программного модуля, описание компонентов и избежать дублирования в хранении информации. Средства управления памятью .NET значительно сокращают количество ошибок в программе ИМ, и тем самым – время на ее отладку.

Библиотека классов ИМ реализована с поддержкой распределенных вычислений, используя технологию .NET Remoting. Модель системы представляется набором процессов, которые в свою очередь могут декомпозироваться посредством подмоделей. При реализации распределенных вычислений на компьютерах в сети производится размещение не отдельных процессов, а подмоделей, такой подход снижает объем данных передаваемый по сети, так как требует меньшего числа синхронизаций данных между процессами.

Диаграмма компонентов ПК приведена на рис. 1. В основе модуля реализации ИМ лежит интегрированная среда разработки Visual Studio .NET, позволяющей расширять ее функциональность путем создания надстроек. Для удовлетворения требования открытости системы, а также для упрощения дальнейшего расширения ПК разработан в виде набора модулей:

- модуль составления формального описания - предназначен для создания формального описания системы в нотации IDEF0, так как эта нотация отражает содержание формального описания ИМ для процессного способа моделирования [5,6];
- модуль реализации ИМ предназначен для создания исходного текста ИМ на основе формального описания и алгоритмов процессов модели. После текст ИМ компилируется и получается скомпилированный файл ИМ, который может быть использован для проведения эксперимента, либо может быть встроен в стороннее ПО;
- модуль отладки ИМ предназначен для проверки правильности реализации алгоритмов процессов и их взаимодействия. Этот модуль позволяет пошагово отлаживать модель и следить за изменением состояния системы;
- модуль реализации анимации ИМ предназначен для создания анимированного представления системы, которое может быть использовано для наглядного рассмотрения протекающих процессов;
- библиотека классов ИМ содержит УПМ и набор базовых классов для реализации ИМ;
- модуль проведения экспериментов предназначен для взаимодействия с подсистемой планирования и проведения экспериментов ПТКИ BelSim2003.

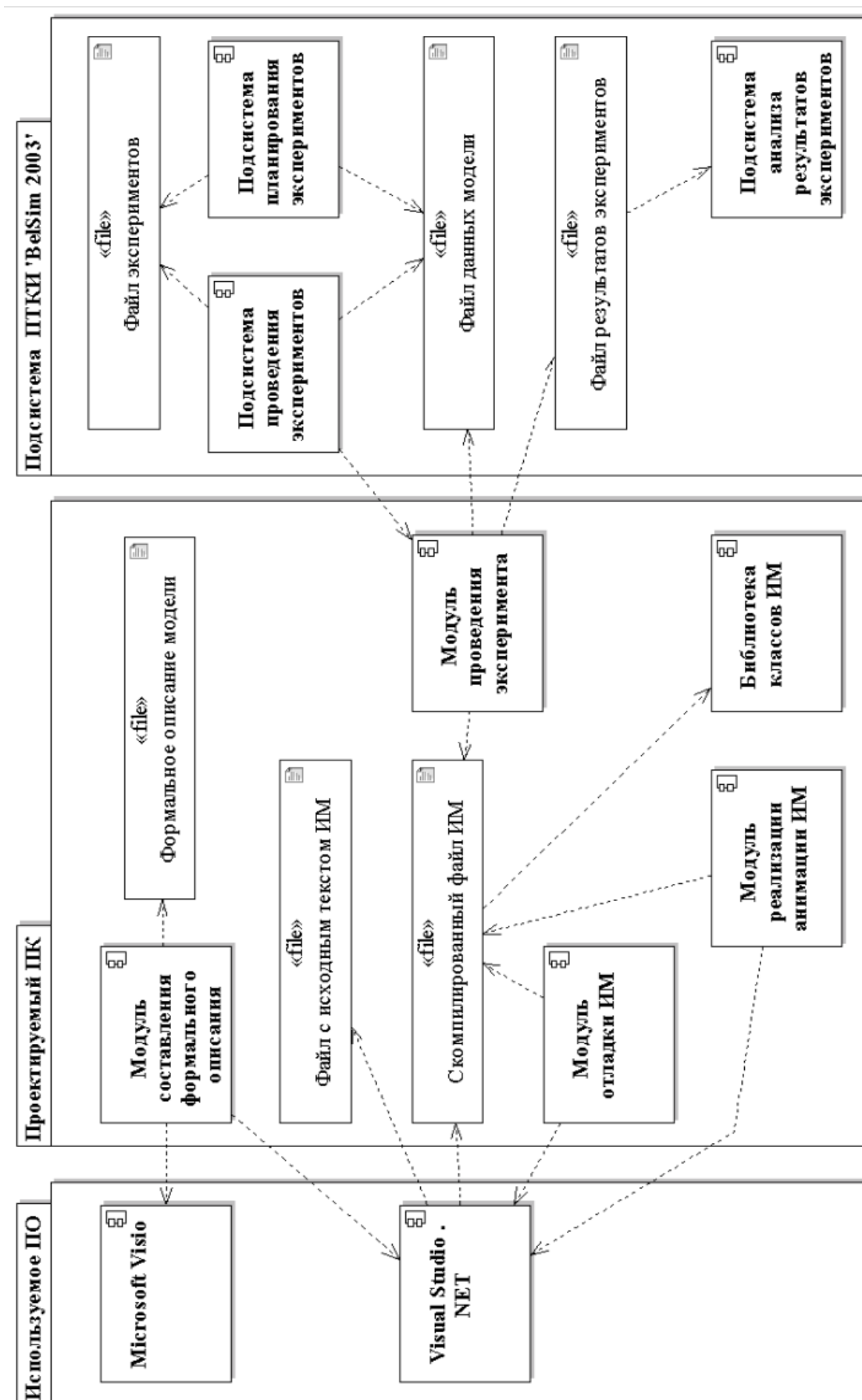


Рис. 1. Диаграмма компонентов ПК с использованием Visual Studio .NET и MS Visio

3. СПОСОБ ОПИСАНИЯ ИМИТАЦИОННОЙ МОДЕЛИ

Модель (Model) представляет собой совокупность взаимосвязанных процессов (Process) и хранилищ данных (Storage), связи которых описываются посредством класса модели. Хранилища данных определяют состояние системы. Процессы описывают функциональные действия компонентов системы и взаимодействуют между собой посредством хранилищ данных. У процесса есть поля (Field), которые служат для соединения с хранилищами данных. Хранилища, с которыми связан процесс, образуют его окружение. К хранилищу может быть присоединено любое количество процессов, в тоже время к полю процесса может быть присоединено только одно хранилище. В процессе своей работы (в течение всего времени моделирования) процесс изменяет состояния хранилищ из своего окружения. Разбиение алгоритма процесса на активности осуществляется путем расстановки операторов синхронизации Wait.

Диаграмма классов, используемых для описания ИМ представлена на рис. 2.

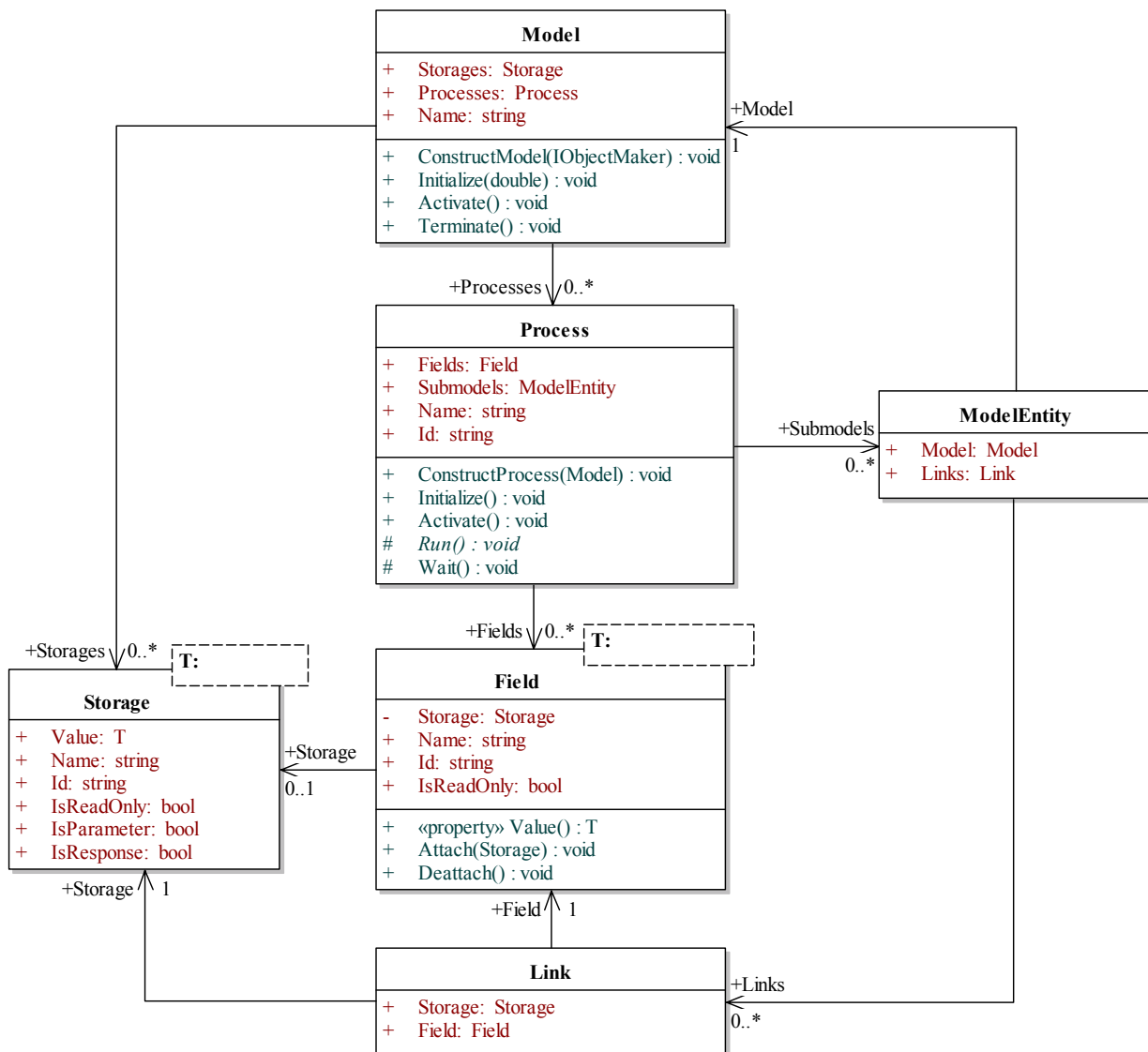


Рис. 2. Диаграмма классов ИМ

Модель характеризуется множествами своих процессов и хранилищ. Любая модель может быть иерархически интегрирована в модель более высокого уровня, для которой она представляется в виде процесса, ничем не отличающегося от других атомарных процессов. Причем подмодель работает в своем модельном времени и для нее используется отдельная УПМ, которая и является алгоритмом процесса, представляющего подмодель с точки зрения модели более высокого уровня. Такая организация позволяет в явном виде учесть тот факт, что между элементами подмодели существуют более тесные и интенсивные взаимодействия, чем между элементами более высокого уровня.

Формальное представление рассмотренного способа описания ИМ:

$$P = (M , \{ F_i \} , \{ W_j \} , \{ L_k = (F_a , S_{Md}) \}), \\ M = (\{ P_i \} , \{ S_j \} , \{ Z_k = (F_{Pab} , S_c) \}),$$

где P – процесс;

F_i – поле процесса;

W_i – подмодель реализующая алгоритм процесса (ModelEntity);

L_a – связь (Link) соединяющая поле F_a и хранилище модели S_{Md} ;

M – модель;

P_i – процесс модели;

S_j – хранилище модели;

Z_k – связь между полем процесса F_{Pab} и хранилищем модели S_c .

4. РЕАЛИЗАЦИЯ УПРАВЛЯЮЩЕЙ ПРОГРАММЫ МОДЕЛИРОВАНИЯ

В результате анализа различных методов моделирования [6], а также на основе результатов проектирования и исследования производственных процессов [7] для увеличения скорости разработки ИМ в качестве метода моделирования используется процессный метод. Этот метод позволяет объединить этапы формализации и создания описания ИМ, что позволяет существенно ускорить процесс проектирования. Модель представляется в виде набора активностей (процессов), которые реализуют процессы, происходящие в моделируемой системе. Предложенный способ записи алгоритма процесса позволяет упростить описание процесса, так как не требует выделения блоков операций в виде отдельных процедур и введения флагов состояния для переходов, которые снижают наглядность описания. Синхронизация процессов осуществляется УПМ с помощью модельного времени T следующим образом. Начальным значением модельного времени принимается минимальное время активизации процесса t_i из всех процессов. После этого происходит активизация всех процессов, в порядке убывания приоритетов. Это продолжается до тех пор, пока состояние системы не перестанет изменяться. Необходимость такого поведения обусловлена тем, что работа процессов происходит не непрерывно и параллельно, а дискретно и последовательно. В результате важным является то, в какой последовательности были активизированы процессы и изменено состояние системы. После того как состояние системы перестает изменяться, происходит корректировка модельного времени. В качестве нового T принимается минимальное t_i , но большее текущего T.

В соответствии с разработанным описанием алгоритма процесса и рассмотренной УПМ необходимо чтобы алгоритмы процессов запускались асинхронно и их можно было прерывать в операторе ожидания, а затем продолжать со следующего оператора. Это реализуется с помощью использования потоков ОС и позволяет не только реализовать разработанный механизм записи алгоритмов процессов ИМ, но и максимально использовать ресурсы многопроцессорных компьютеров.

5. ИССЛЕДОВАНИЕ ПРОГРАММНОГО КОМПЛЕКСА

Для проверки правильности и эффективности работы спроектированной УПМ разработана тестовая модель, представляющая собой совокупность однотипных тестирующих процессов, алгоритм которых состоит из операции с фиксированной трудоемкостью и оператора ожидания с некоторым постоянным интервалом времени. Параметрами модели являются трудоемкость операции в активности процесса и количество процессов. Время моделирования (и, соответственно, число активизаций каждого процесса) выбиралось достаточно большим, чтобы обеспечить возможность замера машинного времени работы модели, выступающего в качестве отклика [5]. Тестирующая модель необходима для связи с модулем проведения экспериментов. Тестирующий процесс создает тестируемую подмодель, в которую добавляет заданное количество процессов. После чего выполняет тестируемую подмодель заданное количество раз и проводит замер времени.

Результаты экспериментов на двухпроцессорной ЭВМ представлены на рис. 3, где T_0 - расчетное время выполнения активности; Tr_2 , Ts_2 - среднее время выполнения активности при параллельном и последовательном выполнении процессов соответственно. Последовательная активизация процессов обеспечивалась назначением им приоритетов по номеру процесса в порядке возрастания.

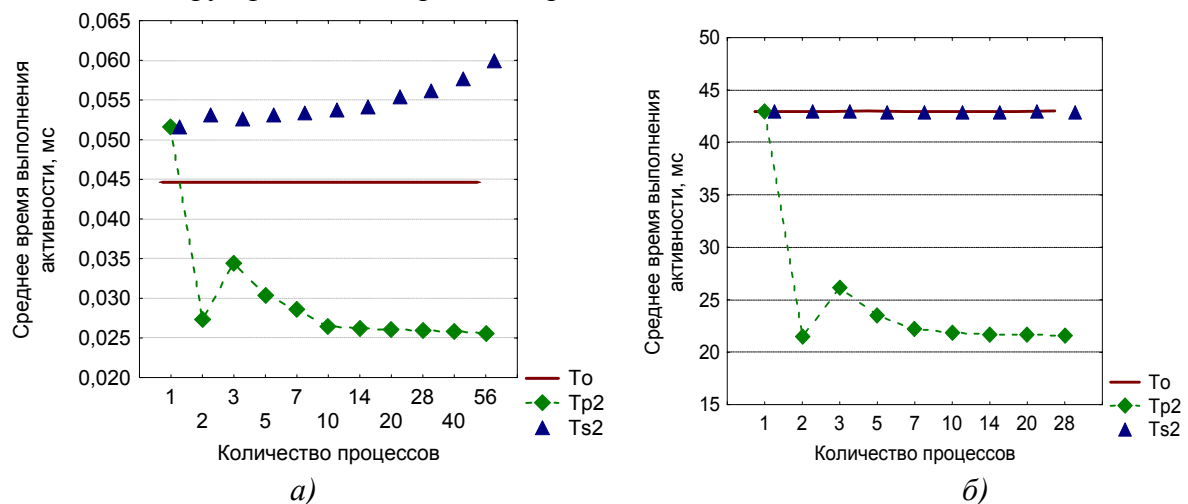


Рис. 3. Графики зависимости времени работы активности от числа процессов.

a – малое время выполнения активности.

б – большое время выполнения активности.

Для двух процессов Tr_2 уменьшается в два раза, при этом, очевидно, алгоритмы процессов выполняются параллельно на разных процессорах. При трех процессах наблюдается некоторый скачок времени выполнения, которое с дальнейшим увеличением количества процессов уменьшается. Такое поведение, вероятно, обусловлено влиянием алгоритма планирования и диспетчеризации потоков выполнения операционной системы, доля времени выполнения которого постепенно начинает снижаться с ростом количества процессов в модели. Рост Ts_2 при увеличении количества процессов обусловлен присутствующей в алгоритме УМП операцией сортировки процессов по приоритету.

Таким образом, путем выполнения алгоритма каждого процесса модели в отдельном потоке появляется возможность эффективно использовать ресурсы многопроцес-

сорных ЭВМ, что особенно актуально при моделировании сложных многоуровневых иерархических систем.

Представленный ПК использован для создания ИМ технологического процесса производства полиэтилентерефталата (ПЭТ) на заводе органического синтеза ОАО «Химволокно» (г. Могилев) с целью решения задачи оптимизации переходного процесса при смене ассортимента. ИМ включает 38 процессов и 2 подмодели и позволяет проводить исследование техпроцесса производства ПЭТ, а также последующую его оптимизацию.

6 ЗАКЛЮЧЕНИЕ

Разработанный ПК является эффективным средством построения ИМ сложных производственных систем, особенно в случае если требуется высокая степень детализации функциональных действий компонентов системы на всех уровнях иерархии. Созданные тестовые ИМ, а также ИМ технологического процесса производства полиэтилентерефталата на заводе органического синтеза ОАО «Химволокно» (г. Могилев), выполненная с целью решения задачи оптимизации переходного процесса при смене ассортимента показывают, что ПК позволяет эффективно создавать ИМ сложных иерархических систем, состоящих из множества процессов. Причем эти ИМ могут быть дополнены и расширены без существенных временных затрат, что достигается минимальной связностью элементов ИМ, которые могут изменяться и декомпозироваться не затрагивая другие элементы. Применение созданного ПК позволяет уменьшить время проведения имитационных экспериментов за счет повышения быстродействия имитационных моделей и использования ресурсов многопроцессорных и многомашинных вычислительных комплексов.

Литература

1. Рыжиков Ю.И. Имитационное моделирование: Теория и технологии. - М.: Альтекс, 2004. - 384 с.: ил.
2. Библиотека электронных ресурсов Московского инженерно-физического института [Электрон. ресурс] / Бородакий, В.Ю. Анализ средств имитационного моделирования распределённых информационных систем. / Окороченко Г.Е. - М., 2007. - Режим доступа: <http://library.mephi.ru/data/scientific-sessions/2007/t12/1-1-12.doc>
3. Петров Р.В. Имитационное моделирование дискретного производственного процесса в ПТКИ BELSIM. / А.И. Якимов, К.В. Захарченко, Р.В. Петров - 41-я студенческая научно-техническая конференция: Материалы конф., 17-21 мая 2005 г. - Могилёв: Белорусско-Российский университет, 2005. - С. 125-126.
4. Петров, Р.В. Имитационное моделирование сложного дискретного производственного процесса на основе ПТКИ BELSIM. / А.И. Якимов, К.В. Захарченко, Р.В. Петров - Известия Гомельского государственного университета имени Ф.Скорины. - №5(32). - 2005. - С. 135-137.
5. Якимов А.И. Имитационное моделирование в ERP-системах управления. / А.И. Якимов, С.А. Альховик - Мн.: Бел. наука, 2005. - 197 с.: ил.
6. Максимей И.В. Задачи и модели исследования операций. Ч. 3. Технология имитации на ЭВМ и принятие решений: Учеб. пособие./И.В. Максимей, В.Д. Левчук, С.П. Жогаль, В.Н. Подобедов - Гомель: БелГУТ, 1999. - 150 с.
7. Петров Р.В. Рациональный выбор параметров производственного процесса на основе ПТКИ BELSIM / А.И. Якимов, К.В. Захарченко, Р.В. Петров - 42-я студенческая научно-техническая конференция: Материалы конф., 16-20 мая 2006 г. - Могилев: Белорусско-Российский университет, 2006. - С. 128.

Петров Роман Владимирович

Магистрант электротехнического факультета
ГУ ВПО Белорусско-Российский университет, г. Могилев
Тел.: +375(222) 25-24-47

E-mail: question@pochta.ru

Альховик Сергей Александрович

Доцент кафедры Автоматизированные системы управления
Белорусско-Российский университет, г. Могилев
Тел.: +375(222) 25-24-47

E-mail: asa@tut.by