

УДК 04.04

А.И. ЯКИМОВ, О.М. ДЕМИДЕНКО, Н.Н. ИВКИНА

АНАЛИЗ МЕТОДОВ ПОСТРОЕНИЯ ИМИТАЦИОННЫХ МОДЕЛЕЙ КОРПОРАТИВНЫХ ИНФОРМАЦИОННЫХ СИСТЕМ

Проведен анализ методов моделирования бизнес-процессов в ERP-системах и предложен метод распределенного построения имитационных моделей с применением информационных технологий, используемых в корпоративных информационных системах промышленных предприятий, от получения исходных данных до принятия решений. Описан разработанный программно-технологический комплекс имитации сложных систем (ПТКИ) BelSim 2, который реализует предложенный метод, используя следующие информационные технологии: IDEF0, UML, C++, ADO, Statistica, MPI, Solver MS Excel.

Ключевые слова: имитационное моделирование; информационные технологии; метод распределенного построения.

ВВЕДЕНИЕ

Основными требованиями к подсистемам имитационного моделирования являются: чтение и запись данных из различных источников; предварительная обработка исходных данных; планирование, проведение и обработка результатов имитационных экспериментов; построение отчетов и графиков. Использование имитационных моделей в составе системы управления промышленным предприятием дополнительно требует создания и выполнения в автоматическом режиме сложных нелинейных сценариев имитационных экспериментов, включающих все этапы от получения исходных данных до принятия решений [1].

Разработанный принцип декомпозиции информационных технологий в комплексной информационной ERP-системе позволяет предложить новый распределенный метод построения и эксплуатации имитационных моделей, в отличие от интегрированных систем имитационного моделирования.

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ МОДЕЛИРОВАНИЯ БИЗНЕС-ПРОЦЕССОВ В ERP-СИСТЕМАХ

Этапы составления содержательного описания, построения концептуальной и формальной моделей в задачах имитационного моделирования до сих пор являются слабо формализованными. В то же время разработан ряд стандартов моделирования бизнес-процессов: примером может служить семейство стандартов IDEF, RUP (компания Rational Software), Causality (компания Computer Associates) [1]. Отраслевыми стандартами являются модели, разрабатываемые государственными и международными общественными организациями (рекомендации ISA, APICS, ISO, TM Forum и др.). При этом целями моделирования являются реинжиниринг бизнес-процессов, автоматизация бизнес-процессов, системные исследования бизнес-процессов и др. Стандарты поддерживаются соответствующими CASE-средствами: BPWin, ErWin, S-Designer, Power Designer, Rational Rose, Design/IDEF, IDEF0/EMTool (белорусско-канадская компания «Ориентсофт») и т.д. На многих промышленных предприятиях применение стандартов группы IDEF является фактическим условием для получения статуса соответствия системы управления качеством требованиям международных стандартов ISO9000, ISO9001.

Наиболее часто используемая IDEF0 методология представляет собой формализованный подход к созданию функциональных моделей – структурных схем изучаемого процесса или системы. Схемы строятся по иерархическому принципу с необходимой степенью подробности и помогают разобраться в том, что происходит в изучаемой системе или процессе, какие функции выполняются и в какие отношения вступают между собой и с окружающей средой ее функциональные блоки. По терминологии, принятой в исследовании процессов, IDEF0-модели относятся к классу концептуальных и являются основой построения имитационных и математических моделей [2]. Технология функционального моделирования систем на основе методологии IDEF0 отработана достаточно хорошо и может быть использована на первых этапах построения имитационной модели в качестве удобного средства представления вербальной и концептуальной модели системы.

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ МОДЕЛИРОВАНИЯ ПРОГРАММНЫХ СИСТЕМ

При моделировании поведения проектируемой или анализируемой системы возникает необходимость не только представить процесс изменения ее состояний, но и детализировать особенности алгоритмической и логической реализации выполняемых системой операций.

Традиционно для этой цели использовались блок-схемы или структурные схемы алгоритмов. Каждая такая схема акцентирует внимание на последовательности выполнения определенных действий или элементарных операций, которые в совокупности приводят к получению желаемого результата.

Наиболее эффективным средством для этого является UML (Unified Modeling Language – унифицированный язык моделирования), который выполняет функции графического описания для объектного моделирования при разработке программного обеспечения. Он обеспечивает коммуникации между заинтересованными сторонами при обсуждении программного проекта с более простым представлением системы. Конструктивное использование языка UML основывается на понимании общих принципов моделирования сложных систем и особенностей процесса объектно-ориентированного анализа и проектирования (ООАП) в частности.

UML обладает возможностью реализации своих конструкций на языках программирования, поддерживающих концепцию объектно-ориентированного программирования (ООП), таких, как, например, C++, Java, Object Pascal. Именно это свойство UML делает его современным средством решения многих задач моделирования сложных систем. Для программной поддержки конструкций языка UML разработаны специальные инструментальные CASE-средства: IBM Rational Rose; Borland Together; Gentleware Poseidon; Microsoft Visio; Telelogic TAU G2; Enterprise Architect (Sparx Systems).

Для моделирования процесса выполнения операций в UML используются так называемые диаграммы деятельности. В контексте UML деятельность представляет собой некоторую совокупность отдельных вычислений. Графически диаграмма деятельности представляется в форме графа деятельности, вершинами которого являются состояния действия, а дугами – переходы от одного состояния действия к другому. Диаграммы деятельности играют важную роль в понимании процессов реализации алгоритмов выполнения операций классов и потоков управления в моделируемой системе. Используемые для этой цели традиционные блок-схемы алгоритмов обладают

серьезными ограничениями в представлении параллельных процессов и их синхронизации [3].

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ ДОСТУПА К ДАННЫМ ИНФОРМАЦИОННОЙ СИСТЕМЫ

Универсальный механизм доступа к данным (Universal Data Access) представляет стратегию обеспечения доступа к любому типу информации предприятия. Он обеспечивает высокопроизводительный доступ к различным источникам информации с помощью единой модели доступа к данным (включая реляционные и нереляционные данные), в том числе к данным, хранящимся на мэйнфреймах, данным электронной почты и файловой системы, текстовым, графическим и географическим данным и др. Для многих современных приложений, использующих данные, характерно подобное разнообразие их источников.

Можно указать следующие наиболее популярные универсальные механизмы доступа к данным: Java Database Connectivity (JDBC); Open Database Connectivity (ODBC); OLE DB/ActiveX Data Objects (ADO).

JDBC – это интерфейс доступа к базам данных, входящий в состав платформы Java. Он позволяет использовать драйверы, написанные на языке Java и работающие на уровне API (поддерживаются СУБД Oracle, Sybase, Informix, DB2 и т.п.), и JDBC-ODBC Bridge для использования существующих ODBC-драйверов.

ODBC (Open Database Connectivity) – это набор интерфейсов для доступа к реляционным данным, удовлетворяющий стандарту CLI (Call Level Interface) ANSI/ISO, разработанный компанией Microsoft. Для доступа к данным конкретной СУБД с помощью ODBC, кроме собственно клиентской части этой СУБД, нужен ODBC Administrator (приложение, позволяющее определить, какие источники данных доступны для данного компьютера с помощью ODBC, и описать новые источники данных), и ODBC-драйвер для доступа к этой СУБД.

Microsoft ActiveX Data Objects (ADO) – это набор библиотек, содержащих COM-объекты, реализующие прикладной программный интерфейс для доступа к данным и используемые в клиентских приложениях. ADO базируется на библиотеках OLE DB, предоставляющих низкоуровневый интерфейс для доступа к данным. OLE DB предоставляет доступ к данным с помощью COM-интерфейсов. Основное назначение ADO – обеспечение простого универсального механизма доступа к данным.

OLE DB – это низкоуровневый интерфейс для доступа к данным. ADO использует OLE DB, но можно использовать OLE DB и напрямую, минуя ADO;

ADO.NET – это следующий шаг в эволюции универсальных механизмов доступа к данным, ориентированный на поддержку распределенных приложений, в том числе использующих мобильные устройства в качестве клиентских рабочих мест. Типичная модель подобного приложения основана на применении кэшируемых наборов данных, представляющих собой хранящуюся в оперативной памяти базу данных, содержащую таблицы, связи между ними, индексы, ограничения и т.п. В ADO.NET (по сравнению с ADO) поддержка подобных наборов данных существенно расширена.

ADO Multi-Dimensional Extensions (ADOMD) – это набор объектов, позволяющих использовать многомерные данные в ADO-приложениях. Такие данные управляются OLAP-серверами (OLAP, Online Analytical Processing), такими, как Microsoft OLAP Server, входящий в комплект поставки Microsoft SQL Server 7.0 (или Analytical Services в

Microsoft SQL Server 2000). OLAP-серверы широко применяются в системах принятия решений, где требуется статистический анализ больших объемов данных [4].

ПРОГРАММНЫЕ СРЕДСТВА СТАТИСТИЧЕСКОЙ ОБРАБОТКИ ДАННЫХ

Одним из обязательных этапов любого научного исследования является статистический анализ данных. Продолжительное время анализ статистических данных был уделом специалистов, так как это требовало серьезной предварительной подготовки. С появлением и совершенствованием современных программ обработки данных статистическая обработка поднялась на новый уровень. Теперь исследователь может и не иметь математической подготовки. Достаточно оперировать статистическими понятиями и, самое главное, правильно выбрать метод анализа. Все осуществимо благодаря компьютеру и новейшим программам.

Одним из наиболее часто используемых методов остается MS Excel – это электронная таблица с достаточно мощными математическими возможностями, где некоторые статистические функции являются просто дополнительными встроенными формулами. В MS Excel невозможно построить качественные научные графики, однако MS Excel хорошо подходит для накопления данных, промежуточного преобразования, предварительных статистических результатов, для построения некоторых видов диаграмм. Существует макрос-дополнение XLSTAT-Pro (<http://www.xlstat.com/>) для MS Excel который включает в себя более 50 статистических функций. Наиболее часто используемыми среди них являются следующие:

STADIA – программа российских разработчиков. Включает в себя все необходимые статистические функции (<http://www.protein.bio.msu.ru/~akula/index.htm>).

SPSS (Statistical Package for Social Science) – самый часто используемый пакет статистической обработки данных (<http://www.spss.com/>). Отличается гибкостью, мощностью и применим для всех видов статистических расчетов.

STATA – профессиональный статистический программный пакет (<http://www.stata.com/>). Один из самых популярных в образовательных и научных учреждениях США наряду с SPSS.

STATISTICA – пакет программ фирмы StatSoft Inc. (США) (<http://www.statsoft.com/>). STATISTICA включает большое количество методов статистического анализа (более 250 встроенных функций), объединенных следующими специализированными статистическими модулями: основные статистики и таблицы, непараметрическая статистика, дисперсионный анализ, множественная регрессия, нелинейное оценивание, анализ временных рядов и прогнозирование, кластерный анализ, факторный анализ, дискриминантный функциональный анализ и др.

JMP – один из мировых лидеров в анализе данных. Развивает статистический пакет SAS Institute (<http://www.jmp.com/>), который выкупил в конце 2002 года известную статистическую программу StatView.

SYSTAT – статистическая система для персональных компьютеров (<http://systat.com/>). Обладает интуитивно понятным интерфейсом. Компания Systat Software также разрабатывает SigmaStat и SigmaPlot, которые являются соответственно программой статистической обработки и программой построения диаграмм. При совместной работе становятся единым пакетом для статистической обработки и визуализации данных.

NCSS – программа, рассчитанная на непрофессионалов в области статистической обработки (<http://www.ncss.com/>).

MINITAB 14 – программный пакет, имеющий хороший интерфейс пользователя, возможности по визуализации результатов работы (<http://www.minitab.com/>).

STATGRAPHICS PLUS – статистическая программа, содержащая более 250 статистических функций, генерирует понятные, настраиваемые отчеты (<http://www.statgraphics.com/>).

На этапе эксплуатации имитационной модели при статистической обработке данных для извлечения дополнительной информации могут быть использованы и современные подходы. Перспективным является, например, анализ сингулярного спектра последовательностей числовых данных [5].

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ ОБМЕНА ДАННЫМИ МЕЖДУ ПРОГРАММНЫМИ СИСТЕМАМИ

Стандартные документы, которые могут содержать текст и ссылки на графические изображения и внешние ресурсы; структурированные документы или записи, такие, как формы HTML; объекты, содержащие данные и методы, например, объекты Java или элементы ActiveX; записи баз данных, которые могут быть представлены на web-странице, и другие документы можно создавать при помощи XML (Exchange Message Language), для отображения содержания применяя HTML и Dynamic HTML (DHTML). XML также

позволяет переназначать, переопределять и отображать содержание из одного источника при помощи других механизмов отображения. Например, хранить единую базу данных на сервере и отображать выделенные данные на нескольких разных устройствах.

Принципиально важным свойством языка XML, обеспечивающим новые функциональные возможности среды web, является его расширяемость. Достижение расширяемости XML основано на двух факторах. Прежде всего, он представляет собой язык метауровня, подмножество известного языка SGML, а не конкретный язык, подобный HTML. Благодаря этому XML выполняет функции языка определения данных. Используя его синтаксис, можно определять различные типы элементов, экземпляры которых образуют содержание конкретных XML-документов, и вводить тем самым адекватный потребностям набор тегов разметки документов. Второй фактор – это использование пространств имен – именованных множеств символов, используемых в качестве имен типов элементов и атрибутов элементов XML-документов. Пространство имен позволяет также явным или неявным образом ассоциировать нужную семантику с именуемыми элементами документов, их атрибутами и допустимыми для них значениями.

Хотя язык XML и базирующаяся на нем платформа стандартов W3C создавались как средство представления информационных ресурсов web, они тем не менее уже находят значительно более широкие применения в различных областях информационных технологий.

При оценке перспектив языка XML нельзя также не учитывать, что он начинает играть существенную роль в других широко распространенных технологиях – CASE-технологиях, технологиях хранилищ данных, потоков работ, в технологиях баз данных, становится основой интеграции информационных ресурсов web и реляционных баз данных. Предпринимаются также шаги, направленные на интеграцию XML-среды с объектными средами и имитационным моделированием [6].

ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ (ООП)

ООП – это набор концепций и идей, позволяющих осмыслить задачу, стоящую при разработке компьютерной программы, а затем найти путь к ее решению более понятным, а, значит, и более эффективным способом. Ключом к пониманию объектно-ориентированных систем является знание различий между классами, объектами и интерфейсами, а также умение применить эти концепции для получения эффективных решений. Качество объектноориентированных решений зависит и от разумной реализации трех принципов ООП: инкапсуляции, наследования и полиморфизма.

В ООП наследованием называют возможность при описании класса указывать на его происхождение от другого класса, что позволяет создать новый класс, в основу которого положен существующий. В полученный таким образом класс можно внести свои изменения, а затем создать новые объекты данного типа. Полиморфизм – это функциональная возможность, позволяющая старому коду вызвать новый. Это свойство ООП дает возможность расширять и совершенствовать программную систему, не затрагивая существующий код.

В рамках рассматриваемых проблем наиболее эффективными и часто используемыми являются: C#, C++, Java, Delphi, Eiffel, Simula, D (язык программирования), Io, Objective-C, Object Pascal, VB.NET, Visual DataFlex, Perl, Php, PowerBuilder, Python, Scala, ActionScript 3.0, JavaScript, JScript.NET, Ruby, Smalltalk, Ada, Xbase++, X++, Vala.

Не обязательно, чтобы один язык был лучше другого, так как они могут обладать различными (непересекающимися) возможностями. Здесь важно не то, какой гаммой средств обладает язык, а насколько они достаточны для поддержания желаемого стиля программирования в области приложений [7].

ИНТЕГРИРОВАННЫЕ СРЕДЫ РАЗРАБОТКИ ПРОГРАММНЫХ СИСТЕМ

Интегрированная среда разработки программного обеспечения (англ. IDE, Integrated Development Environment) – система программных средств, используемая программистами для разработки программного обеспечения. Обычно среда разработки включает в себя текстовый редактор, компилятор и/или интерпретатор, средства автоматизации сборки и отладчик. Иногда также содержит систему управления версиями и разнообразные инструменты для упрощения конструирования графического интерфейса пользователя. Многие современные среды разработки также включают браузер классов, инспектор объектов и диаграмму иерархии классов – для использования при объектно-ориентированной разработке ПО. Хотя и существуют среды разработки, предназначенные для нескольких языков, такие, как Eclipse или Microsoft Visual Studio, обычно среда разработки предназначается для одного определенного языка программирования – как, например, Visual Basic. Примеры сред разработки – Sun Studio, Turbo Pascal, Borland C++, GNU toolchain, DrPython, Borland Delphi, Dev-C++, Lazarus, KDevelop, QDevelop.

Eclipse – это расширяемая, открытая интегрированная среда разработки. Совместно с JDT платформа Eclipse предоставляют множество различных возможностей: подсветка синтаксиса в редакторе, компиляция кода, отладчик уровня исходного кода, навигатор по классам, файловый менеджер и менеджер проектов, интерфейсы для стандартных контролируемых систем исходного кода, рефакторинг кода (<http://www.refactoring.com/>), автоматическое обновление и сборка кода (посредством Update Manager), список задач, поддержка возможности тестирования модулей с помощью JUnit (<http://www.junit.org/>), а также интеграция с инструментом сборки приложений Jakarta Ant

(<http://jakarta.apache.org/ant/index.html>). Особенность Eclipse – это абсолютная нейтральность относительно платформы и языка программирования. На сегодняшний день существуют реализации следующих популярных языков программирования: Java, C/C++, Cobol, Python, Eiffel, PHP, Ruby и C# [8].

Интегрированная среда разработки приложений Geany имеет следующую функциональность: редактор с подсветкой синтаксиса основных языков (C/C++, Java, Perl, PHP, XML); список открытых файлов (вкладок) для редактирования; поддержка разных типов окончания строки и кодировок; автоматическое дополнение в XML-тегах, подсветка синтаксиса множества языков программирования, скриптинга, разметки и даже запросов: ASM, C, C++, CAML, CSS, D, Diff, Fortran, Haskell, X(HTML), Java(Script), LaTeX, Lua, Make, Pascal, Perl, PHP, Python, Ruby, Sh, SQL, TCL, XML и др.; списки символов функций, классов, объектов; поддержка построения проекта; поддержка встроенных и созданных пользователем шаблонов; менеджер проектов; интерфейс для написания модулей к Geany [9].

Среда разработки программного обеспечения NetBeans IDE – бесплатная интегрированная среда разработки приложений (IDE) на языке программирования Java, Ruby и ряде других. Проект NetBeans IDE поддерживается и спонсируется фирмой Sun Microsystems, однако разработка NetBeans ведется независимо сообществом разработчиков (NetBeans Community) и компанией NetBeans Org.

По качеству и возможностям последние версии NetBeans IDE не уступают лучшим коммерческим (платным) интегрированным средам разработки для языка Java, таким, как IntelliJ IDEA, поддерживая рефакторинг, профилирование, выделение синтаксических конструкций цветом, автодополнение набираемых конструкций на лету, множество predefined шаблонов кода и др. В версии NetBeans IDE 6.0 декларируется поддержка UML. NetBeans IDE поддерживает плагины, позволяя разработчикам расширять возможности среды [10].

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ РАСПРЕДЕЛЕННЫХ ВЫЧИСЛЕНИЙ

Существует несколько технологий реализации параллельных вычислений: (N)UMA, DSM, PVM и MPI – все это различные схемы параллельных вычислений. Некоторые из них уже реализованы аппаратно, другие только в программном, а некоторые – и в том, и в другом виде.

(N)UMA: здесь машины пользуются разделяемым доступом к памяти, в которой они могут выполнять свои программы. В ядре Linux реализована поддержка (N)UMA, позволяющая получать доступ к разным областям памяти. При этом задача ядра – использовать ту память, которая находится ближе к процессору, работающему с ней.

DSM уже реализована не только в программном виде, но и в аппаратном. Основная концепция DSM в организации абстрактного слоя для физически распределенной памяти.

История создания кластеров из обыкновенных персональных компьютеров во многом обязана проекту Parallel Virtual Machine (PVM). PVM работает как пользовательская программа, поэтому никаких изменений в ядро системы вносить не нужно

MPI – это открытая спецификация библиотеки передачи сообщений. Самой популярной реализацией MPI является MPICH. Второй по популярности после MPICH можно назвать LAM, также являющейся свободной реализацией MPI.

MOSIX, openMosix, Kerrighed, OpenSSI – полнофункциональные кластерные среды, встроенные в ядро, автоматически распределяющие задачи между однородными узлами.

OpenSSI, openMosix и Kerrighed создают среду единой операционной системы между узлами [11].

Windows Compute Cluster Server 2003 (CCS) разработан для высокотехнологичных приложений, которые требуют кластерных вычислений. Издание разработано для развертывания на множестве компьютеров, которые собираются в кластер для достижения мощностей суперкомпьютера. Каждый кластер на Windows Compute Cluster Server состоит из одного или нескольких управляющих машин, распределяющих задания и нескольких подчиненных машин, выполняющих основную работу. В ноябре 2008 представлен Windows HPC Server 2008, призванный заменить Windows Compute Cluster Server 2003.

Предлагаемый метод распределенного построения имитационных моделей в комплексных информационных ERP-системах состоит в последовательном выполнении следующих этапов.

Этап 1. Построение вербальной модели исследуемого объекта системы, например, с помощью концепции SADT и UML-диаграмм анализа уровней. На этом этапе формируется представление о многоуровневости и иерархичности системы. Содержательное описание автоматизируется с применением информационной технологии IDEF0 и CASE-средства BPWin. Имеется возможность распределения задач проектирования на этом этапе и представления содержания описаний в виде диаграмм, что позволяет говорить уже не о содержательном описании, а о построении вербальной модели. Следует также отметить, что уже на этом этапе за счет его автоматизации обеспечивается контроль за выполнением работ и управлением проектом на ЭВМ, что сокращает длительность реализации проекта (рис. 1).

Основным объектом исследования являются Fi -функции, определяемые на каждом из этапов при построении и эксплуатации имитационной модели ($i = 1, \dots, N$, где N – общее число этапов, $N = 12$). Например, функции $F1, \dots, F3$ на начальных этапах проектирования

ИМ могут быть реализованы в автоматизированном режиме с применением информационных технологий *IDEF*, *DFD*, *RUP* и CASE-средств *BPWin*, *ERWin*, *S-Designer* и др.

Этап 2. Построение концептуальных моделей каждого уровня детализации путем развития функциональных уровней на основе SADT также с применением информационной технологии IDEF0 и CASE-средства BPWin [12]. На этом этапе формируются представления системы для использования в последующем объектно-ориентированного программирования.

Этап 3. Проверка правильности концептуальной модели; если для какого-либо из уровней концептуальная модель не прошла валидацию, имеет место возврат на предыдущий этап. SADT в нотации IDEF0 используется не только для описания системы, но и для изучения и анализа выполняемых ею функций, позволяет документировать результаты исследований, что является необходимым при работе в команде проектировщиков.

Этап 4. Разработка формальной модели системы на основе процессного способа имитации путем составления и описания алгоритмов процессов и отношений между ними с использованием UML-технологии. Решаются вопросы синхронизации процессов и их взаимодействия с управляющей программой моделирования, задания начальных условий, организации сбора статистики и окончания имитации. Выбор процессного

способа имитации обусловлен его универсальным характером, что становится важным при моделировании различных процессов в системе при сохранении особенностей ее структуры.

Этап 5. Построение ИМ системы за счет добавления алгоритмов сбора статистики имитации и построения алгоритмов обработки статистических записей имитации динамики взаимодействия процессов ИМ с применением диаграмм деятельности UML и диаграммы классов. Программным средством поддержки UML-технологии является CASE-средство

Enterprise Architect.

Этап 6. Программирование на каждом уровне детализации ИМ с применением методологии объектно-ориентированного программирования. Это позволяет создавать библиотеку моделей объектов системы, расширять их функциональные возможности. Используется интегрированная система разработки Visual Studio.NET. Система моделирования в ней представлена библиотекой шаблонов. Основным является командный режим проектирования с контролем версий и автоматизированной системой управления проектом.

Этап 7. Верификация программы имитационной модели, состоящая в проверке правильности функционирования программы замыслу исследователя. Разрабатываются тестовые примеры для каждого из компонентов имитационной модели для тестирования программы модели и для контроля за ходом реализации проекта. На этом этапе используется объектно-ориентированное программирование и интегрированная среда разработки Visual Studio.NET.

Этап 8. После успешной верификации программы ИМ и планирования имитационных экспериментов выполняется мониторинг прототипов объекта имитации. Это необходимо для получения исходной информации, реализуется с применением XML-технологии для ввода данных в модель из комплексной информационной системы предприятия и технологии ADO с динамической библиотекой msado15.dll.

Этап 9. Испытание ИМ и технологических характеристик программы модели системы с использованием стандартных пакетов статистической обработки данных имитационных экспериментов Statistica, SPSS.

Этап 10. Проведение серий имитационных экспериментов согласно процедуре МонтеКарло и усреднение значений откликов при каждой комбинации параметров ИМ. Для сокращения времени проведения экспериментов не менее чем в 2,5 раза используется технология распределения вычислений MPI и средство MPICH [13].

Этап 11. Анализ оперативной статистики имитации в пакете Statistica и модификация значений параметров ИМ согласно плану постановки имитационных экспериментов.

Этап 12. Формирование матриц решений, выбор критериев оценки рационального состава параметров ИМ и определение оптимального варианта организации моделируемой системы и рационального состава ресурсов предприятия при эксплуатации имитационной модели. На этом этапе используются программные реализации алгоритмов случайного поиска, в частности, генетический алгоритм [15].

Предлагаемый программно-технологический комплекс имитации сложных систем (ПТКИ) BelSim 2 реализует метод распределенного проектирования имитационных моделей с применением рассмотренных информационных технологий. Например, этапы с первого по девятый используют следующие информационные технологии:

IDEF0 \square^R1 UML \square^R2 C \square^R3 ADO \square^R4 STATISTICA

На этапах проведения исследований ИМ и ее эксплуатации используются информационные технологии, повышающие их эффективность:

STATISTICA \square^R5 MPI \square^R6 STATISTICA \square^R7 Solver MS Excel

В преобразованиях (1) и (2) R_k ($k = 1, \dots, 7$) – отношения между функциями этапов F_i , реализуемых информационными технологиями и соответствующими CASE-средствами.

Например, $R1: \{F1, F2, F3\} \rightarrow \{F4, F5\}$; $R2: \{F4, F5\} \rightarrow \{F6, F7\}$ и др. (рис. 1).

Для автоматизации построения и эксплуатации ИМ могут быть предложены и другие варианты. Для выбора наилучшего из них может быть сформулирована оптимизационная задача.

РЕПОЗИТОРИЙ ГГУ имени Ф. Скорины

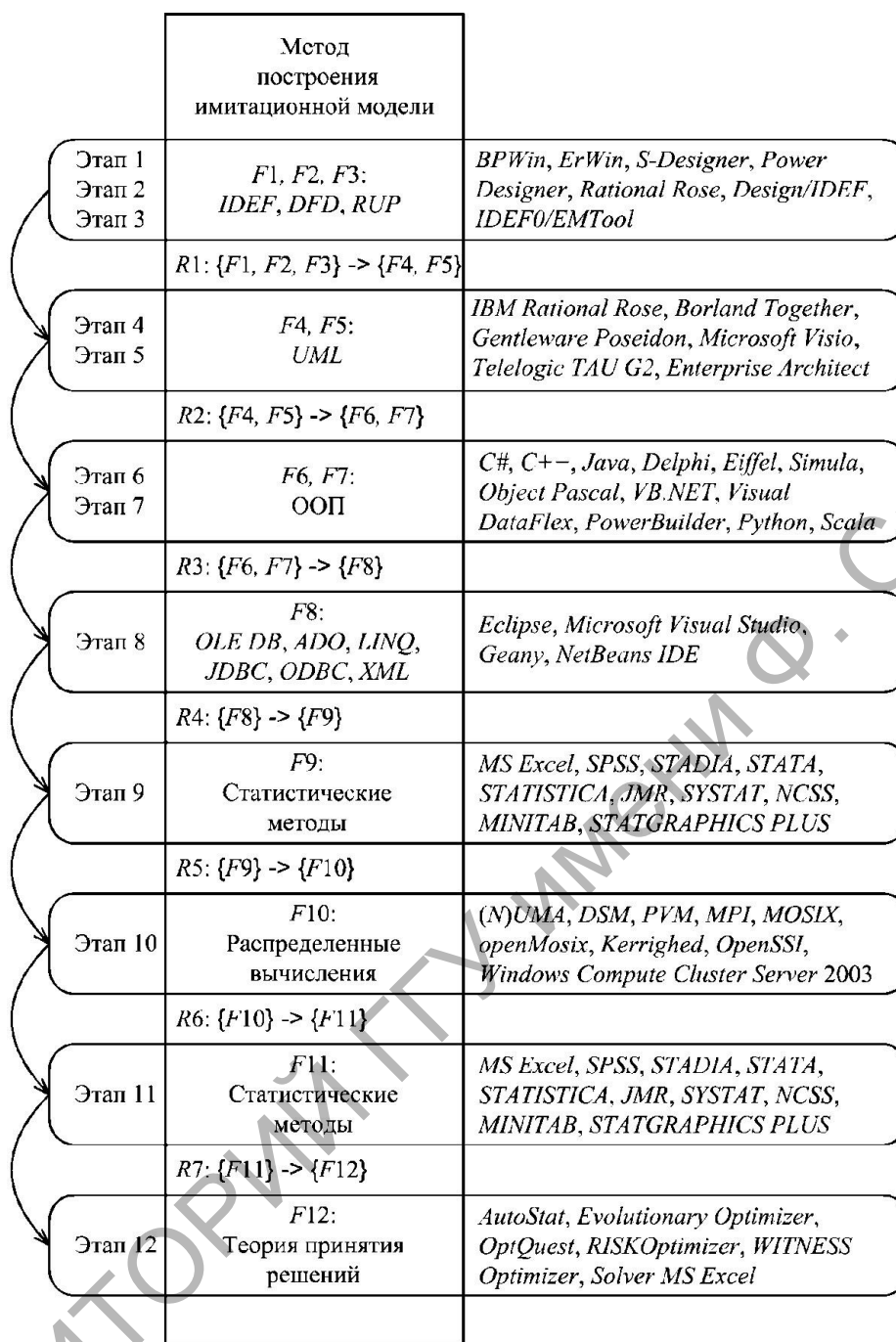


Рисунок 1 – Этапы реализации метода распределенного построения имитационной модели (ИМ)

ЗАКЛЮЧЕНИЕ

Реализация отношений R_k требует разработки оригинальных технологий и средств их поддержки. Например, R_1 представлена технологией построения имитационных моделей на основе IDEF0-диаграмм [12]. На этапе 10 в ПТКИ используется библиотека функций MPI [13], R_6 реализована программным интерфейсом, использующим XML-технологиию для ввода результатов прогона ИМ в пакет STATISTICA [14].

Таким образом, предложено концептуальное развитие актуального научного направления, основанного на универсальном подходе к проблеме имитационного моделирования в автоматизированных системах управления промышленным предприятием.

СПИСОК ЛИТЕРАТУРЫ

1. Якимов А.И., Альховик С.А. Имитационное моделирование в ERP-системах управления. – Мн.: Белорусская наука, 2005. – 198 с.: ил.
2. Рубцов С.В. Опыт использования стандарта IDEF0 // Открытые системы, 2003. – № 1. – С. 53-56.
3. Fowler M. UML distilled: a brief guide to the standard object modeling language. – 3rd ed. – Addison-Wesley Professional, 2003. – 208 p. 4. Троелсен Э. Язык программирования C# 5.0 и платформа .NET 4.5. – 6-е изд. – М.: Вильямс, 2013. – 1312 с.
5. Якимов Е.А., Демиденко О.М., Якимов А.И. Сингулярный спектральный анализ последовательностей данных на этапе эксплуатации имитационной модели // Вестник Брянского государственного технического университета, 2013. – № 1(37). – С. 95-101.
6. Fishwick P.A. Using XML for Simulation Modeling // Proceedings of 2002 Winter Simulation Conference. – San Diego: IEEE Press, 2002. – P. 616-622.
7. Weisfeld M. The Object-Oriented Thought Process. – 4th ed. – Addison-Wesley Professional, 2013. – 336 p.
8. Литвинюк А. Введение в интегрированную среду разработки Eclipse [Электронный ресурс]. – URL: http://www.javaportal.ru/java/ide/intro_eclipse.html.
9. Geany: интегрированная среда разработки приложений [Электронный ресурс]. – URL: <http://www.interface.ru/home.asp?artId=21809>.
10. Монахов В. Язык программирования Java и среда NetBeans. – СПб.: «БХВ-Петербург», 2011. – 704 с.
11. Quesnel F. Sheduling of Large-scale Virtualized Infrastructures: Toward Cooperative Management [Электронный ресурс]. – URL: <https://books.google.by/books>.
12. Альховик С.А., Якимов А.И., Петров Р.В. Технология построения имитационных моделей на основе IDEF0-диаграмм // Математика программных систем: межвузовский сборник научных статей. – Пермь: Пермский государственный университет, 2008. – С. 411.
13. Якимов А.И. Модернизация программно-технологического комплекса имитации сложных систем BelSim для организации распределенных вычислений // Информатика, 2008. – № 2(18). – С. 137-142.
14. Якимов А.И. Технология имитационного моделирования систем управления промышленных предприятий: монография. – Могилев: Белорусско-Российский университет, 2010. – 305 с.
15. Аверченков В.И., Казаков П.В. Эволюционное моделирование и его применение. – Брянск: БГТУ, 2009. – 200 с.

Якимов Анатолий Иванович

ГУ ВПО «Белорусско-Российский университет», Могилев, Беларусь

Кандидат технических наук, доцент кафедры «Автоматизированные системы управления» Тел.:

+375 (222) 252-447, +375 (44) 716-38-16

E-mail: ykm@tut.by

Демиденко Олег Михайлович

УО «Гомельский государственный университет имени Франциска Скорины», Гомель, Беларусь

Доктор технических наук, профессор, проректор по научной работе

Тел.: +375 (232) 603-002, +375 (29) 657-88-63

E-mail: demidenko@gsu.by

Ивкина Наталия Николаевна

ФГБОУ ВПО «Брянский технический университет», Брянск

Аспирант

A.I. YaKIMOV (*Candidate of Engineering Sciences,
Associate Professor of the Department «Automated control systems»
Belarusian-Russian University, Mogilev, Belarus*)

O.M. DEMIDENKO (*Doctor of Engineering Sciences, Professor, Vice Rector for Research) Francysk
Skaryna Homiel State University, Gomel, Belarus*)

N.N. IVKINA (*Post-graduate Student)
Bryansk State Technical University, Bryansk*)

ANALYSIS OF METHODS FOR CONSTRUCTING SIMULATION MODELS OF CORPORATE INFORMATION SYSTEMS

The analysis methods of modeling business processes in ERP systems and a proposed distributed method of building simulation models with application of information technologies used in enterprise information systems of industrial enterprises, from receiving the original data before making decisions. This article describes a software technological complex simulation of complex systems (PTCI) BelSim 2, which implements the proposed method, using the following information technologies: IDEF0, UML, C++, ADO, Statistica, MPI, MS Excel Solver.

Keywords: *imitating modeling; information technology; method distributed build.*

BIBLIOGRAPHY (TRANSLITERATED)

1. Yakimov A.I., Al'xovik S.A. Imitacionnoe modelirovanie v ERP-sistemax upravleniya. – Mn.: Belorusskaya nauka, 2005. – 198 s.: il.
2. Rubcov S.V. Opy't ispol'zovaniya standarta IDEF0 // Otkry'ty'e sistemy', 2003. – № 1. – S. 53-56.
3. Fowler M. UML distilled: a brief guide to the standard object modeling language. – 3rd ed. – AddisonWesley Professional, 2003. – 208 p.
4. Troelsen E'. Yazy'k programirovaniya C# 5.0 i platforma .NET 4.5. – 6-e izd. – M.: Vil'yams, 2013. – 1312 s.
5. Yakimov E.A., Demidenko O.M., Yakimov A.I. Singulyarny'j spektral'ny'j analiz posledovatel'nostej danny'x na e'tape e'kspluatacii imitacionnoj modeli // Vestnik Bryanskogo gosudarstvennogo texnicheskogo universiteta, 2013. – № 1(37). – S. 95-101.
6. Fishwick P.A. Using XML for Simulation Modeling // Proceedings of 2002 Winter Simulation Conference. – San Diego: IEEE Press, 2002. – P. 616-622.
7. Weisfeld M. The Object-Oriented Thought Process. – 4th ed. – Addison-Wesley Professional, 2013. – 336 p.
8. Litvinyuk A. Vvedenie v integrirovannuyu sredu razrabotki Eclipse [E'lektronny'j resurs]. – URL: http://www.javaportal.ru/java/ide/intro_eclipse.html.
9. Geany: integrirovannaya sreda razrabotki prilozhenij [E'lektronny'j resurs]. – URL: <http://www.interface.ru/home.asp?artId =21809>.
10. Monaxov V. Yazy'k programirovaniya Java i sreda NetBeans. – SPb.: «BXV-Peterburg», 2011. – 704 s.
11. Quesnel F. Sheduling of Large-scale Virtualized Infrastructures: Toward Cooperative Management [E'lektronny'j resurs]. – URL: <https://books.google.by/books>.
12. Al'xovik S.A., Yakimov A.I., Petrov R.V. Texnologiya postroeniya imitacionny'x modelej na osnove IDEF0-diagramm // Matematika programmny'x sistem: mezhvuzovskij sbornik nauchny'x statej. – Perm': Permskij gosudarstvenny'j universitet, 2008. – S. 4-11.
13. Yakimov A.I. Modernizaciya programmno-texnologicheskogo kompleksa imitacii slozhny'x sistem BelSim dlya organizacii raspredelenny'x vy'chislenij // Informatika, 2008. – № 2(18). – S. 137-142.

14. Yakimov A.I. Tekhnologiya imitacionnogo modelirovaniya sistem upravleniya promy'shlenny'x predpriyatij: monografiya. – Moglev: Belorussko-Rossijskij universitet, 2010. – 305 s.
15. Averenkov V.I., Kazakov P.V. E'volucionnoe modelirovanie i ego primenenie. – Bryansk: BGTU, 2009. – 200 s.

РЕПОЗИТОРИЙ ГГУ имени Ф. СКОРИНЫ