

Integrating SQL-Based DBMSs with NoSQL Datastores at the Object-Mapping Layer

This paper will discuss and evaluate an approach to building an object-mapping layer that provides integration of multiple different data management systems, including SQL-based DBMSs and popular NoSQL systems. A prototype object-mapping framework was developed to illustrate basic concepts of such integration including unified entity-based query language, transparent object mapping and a high-level interface, that still allows low-level optimizations and delivers sufficient performance for most use cases. Unified query language makes it possible to translate a single query to different underlying datastores without a need to modify the query while object-mapping layer manages object construction and lifecycle, providing simple interface for persisting, modifying and deletion of objects. In conclusion, performance of the implemented framework is measured and compared with popular ORM solution.

Keywords: SQL, NoSQL, polyglot persistence, service-oriented architecture, object-relational mapping, object-document mapping, integration

References

1. Merriman D. *On Distributed Consistency*. 2010. URL: <http://blog.mongodb.org/post/475279604/on-distributed-consistency-part-1>
2. Cattell R. *Scalable SQL and NoSQL Data Stores*. 2011. URL: <http://www.cattell.net/datastores/Datastores.pdf>
3. Aslett M. *NoSQL, NewSQL and Beyond: The drivers and use cases for database alternatives*, 451 Group, 2011. URL: <https://451research.com/report-long?icid=1651>
4. Kuznetsov S. D., Poskonin A. V. Raspredelelnnye gorizonttal'no masshtabiruemye reshenija dlja upravlenija dannymi. *Trudy Instituta sistemnogo programirovanija RAN*. 2013. V. 24 P. 327—358.
5. Stonebraker M., Cetintemel U. "One Size Fits All": An Idea Whose Time Has Come and Gone, ICDE '05. *Proceedings of the 21st International Conference on Data Engineering, Washington*, 2005.
6. Entity-Attribute-Value model. URL: http://en.wikipedia.org/wiki/Entity-Attribute-Value_model
7. Fowler M. *Polyglot Persistence*, 2011. URL: <http://www.martinfowler.com/bliki/PolyglotPersistence.html>
8. Kuznetsov S. D., Poskonin A. V. Vozmozhno li sotrudnichestvo SQL i NoSQL? *Otkrytye sistemy. SUBD*. 2013. N. 9. P. 38—41.
9. Francia S., Hileman J. *Augmenting RDBMS with MongoDB for eCommerce*. URL: <http://www.nosqldatabases.com/main/2011/4/11/augmenting-rdbms-with-mongodb-for-ecommerce.html>
10. Poskonin A. V. Web-prilozhenija i dannye: problemy abstrakcii i sshtabiruемости. *Trudy Instituta sistemnogo programirovanija RAN*. 2012. V. 23. P. 159—171.
11. Miller J. *Design Patterns for Data Persistence*. 2009. URL: <http://msdn.microsoft.com/en-us/magazine/dd569757.aspx>
12. Fowler M. *Patterns of Enterprise Application Architecture*. Boston: Addison Wesley, 2002.

УДК 681.518: 339.13

Э. М. Димов, д-р техн. наук, проф.,

О. Н. Маслов, д-р техн. наук, проф., зав. каф., e-mail: maslov@psati.ru,

Ю. В. Трушин, канд. техн. наук, доц.,

Поволжский государственный университет телекоммуникаций и информатики, г. Самара

Выбор средств программного обеспечения процесса статистического имитационного моделирования

Рассматривается проблема выбора средств программного обеспечения процесса статистического имитационного моделирования (СИМ) сложных систем организационно-технического типа. Представлены примеры реализации СИМ-моделей с применением универсального программного языка Delphi и среды имитационного моделирования AnyLogic.

Ключевые слова: метод статистического имитационного моделирования, средства программного обеспечения, универсальный язык Delphi, среда имитационного моделирования AnyLogic

Введение

Разработка и реализация систем управления (СУ) иерархическими и многокритериальными сложными системами (СС) организационно-технического типа (социально-экономическими, экологи-

ческими, военными и т. п.), элементами которых являются лица, принимающие решения (ЛПР), до настоящего времени является актуальной проблемой, которая имеет важное практическое значение. Решению данной проблемы способствует создание

"имитационных систем" [1, 2], каждая из которых представляет собой человеко-машинную диалоговую СС и состоит из математических (алгоритмических) моделей и системы процедур, объединяющей эти модели с ЛПР, а также специального математического обеспечения, включающего систему алгоритмов для решения конкретных задач, в том числе оптимизационных.

Перспективы повышения эффективности действий ЛПР в составе СУ с помощью статистического имитационного моделирования (СИМ) по версии метода Димова—Маслова (МДМ) рассмотрены в работе [1]. Согласно методике СИМ по МДМ, выбор программного обеспечения (ПО) является одним из важных предварительных этапов реализации СИМ-модели [2]. Помимо стоимости, которая в условиях рынка имеет существенное значение, выбранный вариант ПО должен быть общедоступным и универсальным, приемлемым для ЛПР по эффективности и комфорту использования. Цель статьи — не претендуя на завершенность анализа, дать оценку вариантов ПО, пригодных для реализации СИМ-моделей по МДМ, наиболее распространенных в настоящее время.

Требования и сравнительные возможности средств ПО СИМ

Требования к СИМ-модели (комфортный для ЛПР режим взаимодействия с ней; ввод и вывод информации в удобной форме; возможность модификации и интеграции с другими компонентами СУ и т. д.) во многом определяются требованиями, которые предъявляются к ПО процесса СИМ. Главными из них являются:

- гибкость в смысле возможности моделировать СС с разным уровнем сложности [3];
- наличие хороших средств отладки, которые позволяют отслеживать отдельные объекты по всей модели, чтобы убедиться в правильности их обработки и проверять состояние модели при каждом возникновении нового события;
- дружественный интерфейс и работа с графикой.

Важным также является наличие механизма генерирования независимых случайных численных величин (СЧВ) с исходным равномерным распределением на интервале $[0, 1]$, чтобы, во-первых, с заданным качеством получать в дальнейшем все требуемые законы распределения СЧВ; во-вторых, для каждого прогона СИМ-модели применять одни и те же исходные условия, используя фиксированные наборы СЧВ и перед каждым прогоном приводя статистические счетчики в исходное состояние.

При выборе ПО необходимо решить, для какой платформы оно будет предназначено, поскольку большинство программных средств СИМ работают сегодня под управлением Windows и на рабочих станциях UNIX, но все большее распространение получает операционная система Mac. При возмож-

ности работы пакета на разных платформах ему необходимо обеспечить совместимость с другими платформами и уделить внимание тому, какие операционные системы он поддерживает. Для оценки результатов моделирования в ПО должна быть возможность создавать стандартные отчеты и сохранять отработанные сценарии СИМ в базе данных.

Первые версии ПО, которые с 1955 г. разрабатывались на языках типа FORTRAN [4], уже имели признаки СИМ-моделей: входные переменные в них случайным образом изменялись, а выходные подвергались статистической обработке. С тех пор были созданы методики и инструменты, призванные упростить и ускорить создание СИМ-моделей. В 60-е годы прошлого века появились специализированные языки GPSS, SIMSCRIPT, GASP и SIMULA, а в 1987 г. — графические среды, позволяющие сформировать структуру процесса СИМ путем комбинирования визуальных блоков. Сегодня на рынке присутствует более 50 программных продуктов для разработки СИМ-моделей различных объектов и процессов, которые можно разделить на три группы [5]:

- универсальные инженерные языки программирования;
- специализированные языки имитационного моделирования;
- среды имитационного моделирования.

Универсальные языки программирования BASIC, FORTRAN, C/C++, Pascal, Java и др. [5] являются традиционным средством реализации СИМ-моделей. В основе моделирующего алгоритма здесь лежит технология метода Монте-Карло (ММК), которая использует процедуру "разыгрывания" СЧВ, соответствующих случайным факторам, воздействующим на объект. Алгоритм представляет собой последовательность процедур, которая описывает состояния моделируемой СС в определенные дискретные моменты времени. Процесс СИМ по МДМ при этом заключается в многократном периодическом "вычислении" состояния СС и ее элементов. Достоинством данного подхода является его универсальность как возможность реализации модели СС любой сложности. Вместе с тем, учет особенностей реальных объектов (в том числе бизнес-процессов), а также обилие противоречивых условий их существования, приводят к тому, что разработка СИМ-модели на универсальном языке требует от ЛПР высокой квалификации и значительных временных затрат [4, 6].

Специализированные языки имитационного моделирования реализуют стандартные операции типа "создать объект", "передать объект", "ожидать в течение указанного промежутка времени", "сгенерировать СЧВ с указанным законом распределения", "выполнить операцию" и т. д., которые представляют собой блоки в виде последовательности команд на одном из универсальных языков [4]. Они ком-

пактны и имеют широкий круг приложений, различаются способами учета времени, сложностью изменения структуры модели и способами проведения экспериментов, но все требуют специальной подготовки ЛПП, который должен написать программу в терминах языка для конкретного объекта СИМ. Многие из них основаны на теории систем массового обслуживания (СМО), где поступающие заявки становятся в очередь на обслуживание агрегатами. Основной объект в подобных языках — это пассивный транзакт (заявка на обслуживание), который может представлять собой ресурсы (работников, деталей, сырья), документы, сигналы и т. п. "Перемещаясь" по СИМ-модели, транзакты становятся в очереди к одноканальным и многоканальным агрегатным устройствам, захватывают и освобождают их, расщепляются, уничтожаются и т. п. Данный подход в интересах СИМ поддерживают GPSS/PC, GPSS/H, GPSS World, Object GPSS, Arena, SimProcess, Enterprise Dynamics, AutoMod и др.

Среды имитационного моделирования, в отличие от языков, не требуют программирования в виде последовательности команд или предварительного синтеза алгоритма моделирования, что является их очевидным достоинством. Вместо написания программы ЛПП составляет модель из графических блоков или заполняет специальные бланки (формы) имитатора. Модель при этом составляется путем комбинирования элементов, входящих во встроенную библиотеку. Использование такого подхода позволяет повысить наглядность СИМ, хотя большинство сред являются лишь графическими интерпретациями соответствующих специализированных языков моделирования. В крупные имитационные системы входят средства автоматического анализа результатов и оптимизации процесса СИМ, а также управления проведением экспериментов.

Работа имитатора может быть основана как на независимом языке (среда Arena разработана на языке SIMAN, система MapSys — на языке MapSim), так и на собственном, разработанном специально для данной среды (среда EcosimPro основана на языке EL — EcosimPro Language). В объектно-ориентированных имитаторах поведение отдельных блоков задается посредством написания функции на одном из универсальных языков программирования (в среде AnyLogic используется язык Java, в ProcessModel — BASIC). Недостатком большинства проблемно-ориентированных имитаторов является сложность формализации бизнес-процесса и интерпретации результатов СИМ. Кроме того, в отличие от языков моделирования и программирования, из-за предопределенности элементов СИМ-модели область ее приложения получается существенно ограниченной. Помимо среды AnyLogic на рынке сегодня присутствуют среды ARIS Business Simulator, ReThink, Simulink.

В соответствии с изложенным оптимальный выбор ПО для проведения СИМ представляет собой непростую задачу, практическое решение которой ЛПП не всегда удается найти. Дорогостоящие графические среды не обладают гибкостью, необходимой для моделирования многих бизнес-процессов. В то же время они обеспечивают самую высокую скорость разработки СИМ-модели. Специализированные языки неудобны в использовании и уступают имитаторам по степени наглядности, однако обладают высокой степенью гибкости. Универсальные языки программирования сложны в изучении и требуют значительных временных затрат при разработке СИМ-моделей, зато они позволяют моделировать практически любые СС, создавать, менять и корректировать свойства объектов ПО (меню, элементов управления, окон и т. д.), что позволяет им в целом успешно конкурировать с имитаторами. Рассмотрим конкретные примеры программной реализации СИМ-моделей по МДМ с использованием универсальных языков и сред имитационного моделирования.

Программная реализация на языке Delphi СИМ-модели бизнес-процесса оплаты клиентами услуг телекоммуникационной компании

Сравнительные достоинства языка Delphi обусловлены следующими его особенностями.

1. Компилятор в машинный код со скоростью более 120 тыс. строк в минуту обеспечивает производительность, необходимую и достаточную для построения приложений в архитектуре клиент—сервер; предполагает легкость разработки и быстрого времени проверки готового программного блока при требуемом высоком качестве кода. Delphi обеспечивает разработку без ручного написания кода (хотя это тоже возможно), когда ЛПП выбирает из палитры компонентов готовые элементы и еще до компиляции может видеть результаты своей работы, перемещаться по данным и представлять их в том или ином нужном ему виде. В этом смысле проектирование в Delphi мало отличается от проектирования в интерпретирующей среде, но после компиляции получается код, который выполняется в 10...20 раз быстрее, чем при исполнении с помощью интерпретатора.

2. В объектно-ориентированной модели программных компонентов Delphi основной упор делается на максимальном повторном использовании кода — это позволяет ЛПП быстро строить приложения из заранее подготовленных объектов, а также создавать собственные объекты. Наличие процедуры Randomize, входящей в пакет стандартных библиотек Delphi, позволяет инициализировать генератор СЧВ. Ограничений по типам создаваемых объектов не существует, разработка интерфейса в Delphi является приемлемо простой задачей для ЛПП.

3. Клиент-серверная версия Delphi позволяет реализовать высокопроизводительные приложения при работе с разными источниками информации, обеспечивает прозрачность подключения новых механизмов доступа к данным.

При использовании СИМ-модели бизнес-процесса оплаты клиентами услуг крупной региональной телекоммуникационной компании [1, 2] после запуска приложения ЛПР видит на экране ЭВМ окно безопасности, в поле которого следует ввести пароль доступа. Затем нажатием кнопки "Войти" окна или клавиши "Enter" стандартной клавиатуры он получает доступ к основному окну программы (рис. 1), которое содержит главное меню, панель управления функционалом программы и поля для ввода входных параметров СИМ-модели.

Исходные данные разделены по категориям, каждой из которых соответствует отдельная закладка окна, чтобы ЛПР было легче ориентироваться в многочисленных параметрах СИМ-модели. В левой части главного окна программы расположены кнопки, дублирующие команды главного меню. Панель разделена тремя областями, которые активируются (раскрываются) или сворачиваются, в зави-

симости от выполняемых действий. Так, например, на этапе ввода исходных данных ЛПР не нужны элементы управления результатами СИМ и отчетами, поскольку они еще не получены. Все кнопки панели управления СИМ-модели имеют соответствующие клавишные комбинации, информацию о которых, а также другие справочные данные можно получить, вызвав из главного меню справку по программе.

Завершается моделирование появлением окна результатов СИМ (рис. 2), где в удобном для ЛПР виде отображаются выходные данные модели, которые он при необходимости (переключившись на вкладку "Графики" окна) может увидеть в графическом формате — как диаграммы. Предусмотрены печать таблицы выходных данных моделирования и графиков непосредственно из окна результатов, а также отправка результатов в виде отчета в приложение MS Excel, где можно просматривать и анализировать данные средствами этого табличного процессора. Цветовая гамма приложения выдержана в стандартных Windows-тонах, расположение меню и элементы управления соответствуют стандартам данной операционной системы, что способствует быстрому освоению программы ЛПР.

Программная реализация на языке Delphi СИМ-модели бизнес-процесса реанимирования нефтяных скважин

Данная СИМ-модель также является многооконным Windows-приложением, оснащенным главным меню, средствами управления окнами и средствами исследования данных. Импорт исходных данных осуществляется в одноименном окне, открытие которого реализуется выбором команды главного меню "Исходные данные → Импортировать" или нажатием комбинации клавиш Ctrl + I. Загруженные в модель данные (подробный перечень см. в работе [1]) имеют табличное представление и при необходимости могут быть отредактированы вручную или дополнены на выбор ЛПР, для чего предусмотрены соответствующие элементы управления. После того, как данные загружены и подготовлены, запуск СИМ реализуется командой "Моделирование → Запустить" главного меню или нажатием клавиш Ctrl + M.

По завершении СИМ в отдельном окне (аналогично рис. 2) представляется таблица с результатами реанимирования скважин, которая содержит выходные характеристики бизнес-процесса:

- наиболее эффективный вариант реанимирования;
- новое число качаний;
- наиболее подходящий вид промывочной жидкости;
- оптимальное время и вероятность успешности реанимирования для каждой скважины [1].

При нажатии клавишей мыши на соответствующую строку в таблице на экран выводится окно,

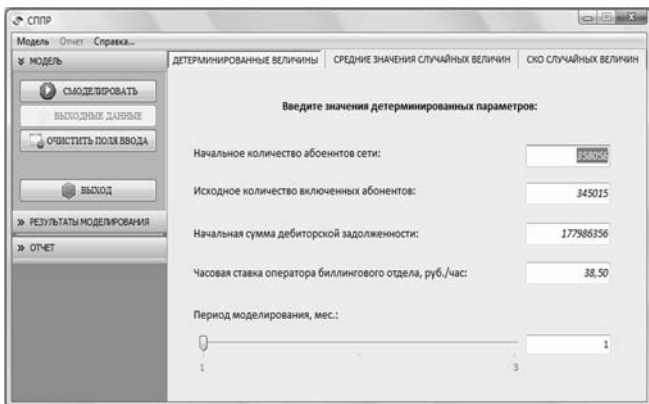


Рис. 1. Основное окно программы (СИМ-модели)

ЧИСЛО МЕСЯЦА	СТАТУС	ВСЕГО АБОНЕНТОВ	ВКЛЮЧЕНО АБОНЕНТОВ	ОБЩАЯ СУММА НАЧИСЛЕНИЙ	КОЛИЧЕСТВО ОПЛАТИВШИХ	КОЛИЧЕСТВО ОПЛАТИВШИХ ОТКЛЮЧЕННЫХ
1	-	358065	345019	5098983,51	6270	878
2	-	358080	345318	4396164,94	6187	866
3	-	358100	345616	2813742,19	6264	877
4	-	358133	345961	4411781,43	6253	875
5	-	358156	346333	5835442,74	6173	864
6	-	358183	346712	4990545,88	6205	869
7	-	358202	347170	5622295,72	6199	868
8	-	358210	347599	3815705,05	6268	251
9	-	358233	348050	3448932,32	6229	249
10	-	358261	348490	3181486,39	6205	248

Среднее число абонентов периода:	362899	Количество выключенных абонентов за период:	209213
Количество подключающихся за период абонентов:	1052	Общее количество времени, затраченного на перерасчет абонентов, час.:	325,9
Количество расторгнутых за период договоров по инициативе абонента:	178	Затраты, связанные с перерасчетом абонентов:	12550,15
Количество расторгнутых за период договоров по инициативе компании:	3	Общая сумма начислений за период:	330781247,88
Количество составленных претензий за период:	76	Общая сумма оплаты за период:	381902859,87
Процент оплаты по претензиям за период, %:	0,66	Средняя величина дебиторской задолженности периода:	172436067,84 руб.
Количество выключенных абонентов за период:	159703		

Рис. 2. Фрагмент окна результатов СИМ

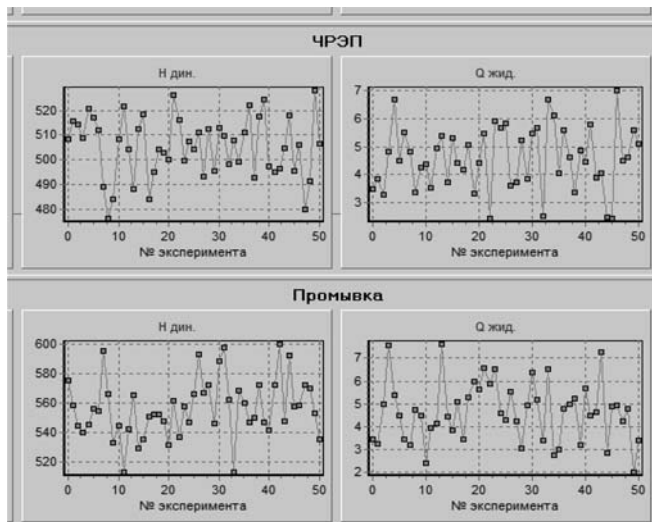


Рис. 3. Детализированные результаты СИМ (фрагмент вкладки "Графики")

содержащее детализированную информацию по заданной скважине: на вкладке "Расчеты" — статический и динамический уровень, дебит жидкости и нефти после реанимирования, предполагаемая стоимость смоделированных работ, общая стоимость работ и общее эффективное время мероприятий; на вкладке "Графики" — изменение моделируемых величин в форме линейных графиков (фрагмент данной вкладки в увеличенном виде см. на рис. 3).

Команда "План-графики → Сформировать" на основе данных СИМ позволяет автоматически формировать план-графики бригадам по реанимированию, содержащие: номер бригады, номер скважины, вариант реанимирования и его ожидаемые результаты (рис. 4), а также информацию об ответственном исполнителе, отметки о принятии к исполнению и, по завершении работ, дату исполнения — последние два поля автоматически обновляются при каждом открытии план-графика в СИМ-модели.

№ бригады	№ скважины	Вариант реаним.	Изменение йкал.	Прон. жидкость	Предв. дата реанимирования, дд.мм	Время реанимирования, часы
524	6017	Шоке	3	-	01/04	10:00
723	5864	Шоке	4	-	07/04	08:00
115	1224	РЭП-промывка	3	Пр.в. нмл-вз	07/04	12:30
371	3469	РЭП-промывка	3	Пл.в. нмл-вз	12/04	14:00
248	2548	Шоке	2	-	06/04	11:30
319	3859	РЭП	4	-	01/04	10:00
422	10587	РЭП	5	-	04/04	08:00
574	4586	Шоке	2	-	05/04	13:00
108	5896	РЭП-промывка	3	Нефть	11/04	12:30
267	10248	Шоке	3	-	06/04	10:00
284	9258	РЭП-промывка	3	Прес. вода	04/04	11:00
527	6584	промывка	-	Горч. нефть	03/04	12:00

Редактировать план | Параметры план-графика | Расстояние до скважины от цеха, км.: 3,2 | Стоимость работ, руб.: 20484,23 | Упущенная выгода, руб./сут.: 52815,87 | Сохранить | Отправить исполнителю (по e-mail) | Экспорт в Excel | Распечатать | Выполнить

Рис. 4. Формируемый СИМ-моделью план-график бригадам по реанимированию нефтяных скважин

Сформированный план-график может быть сохранен, распечатан, отправлен исполнителям по электронной почте или экспортирован в MS Excel. Пользовательский интерфейс включает набор необходимых ЛПР элементов управления (меню, кнопки и т. д.), он прост и не избыточен, скрывает промежуточные вычисления и преобразования, реализован в стандартном Windows-оформлении и внешне соответствует интерфейсу корпоративной информационной системы, что делает его привычным и понятным специалисту-нефтянику.

Программная реализация в среде AnyLogic СИМ-модели бизнес-процесса "Выполнение заказа"

Среда AnyLogic, которая является одной из немногих отечественных разработок, получивших признание за рубежом [4], предназначена для графического создания СИМ-моделей с использованием объектно-ориентированного языка программирования Java. Она поддерживает три методики моделирования: системную динамику, дискретно-событийное и агентное моделирование, позволяя комбинировать эти подходы при разработке СИМ-модели. Разработчики AnyLogic непрерывно совершенствуют свой продукт, регулярно выпускают его новые версии и пополняют библиотеки готовых компонентов.

В рассматриваемом случае необходимо, во-первых, синтезировать структуру бизнес-процесса с учетом логики его функционирования, чтобы каждый блок бизнес-процесса, содержащий СЧВ, был отражен на схеме, показанной в верхней части рис. 5. Напомним, что каждый "прогон" СИМ-модели представляет собой последовательное моделирование всех СЧВ бизнес-процесса с учетом его логики в течение заданного периода моделирования [2; 7]. Показанная на рис. 5 схема выглядит следующим образом:

- *newClient* — начало бизнес-процесса, поступающие в СМО заявки (время между поступающими заявками — СЧВ);
- *queueToService* — очередь (существующая в общем случае) на оформление заказа. Задается ее вместимость, а также время (тайм-аут), по истечении которого не дождавшаяся обслуживания заявка покидает СМО (*rejectedClients*);
- *service* — этап оформления заказа, выполняемый отделом сбыта (длительность оформления заказа представляет собой СЧВ).

Элемент *rejectOrder* представляет возможный отказ клиента от заказа, это тоже СЧВ. "Отказавшиеся" клиенты накапливаются в блоке *rejectedClients*. Если заявка проходит дальше, она попадает в очередь на планирование производства (*queueToPlanning*), а затем — непосредственно в объект *planning* (длительность планирования производства, СЧВ).

Возможная нехватка запасов (СЧВ) реализуется с помощью элемента *noResources*. Если ресурсов

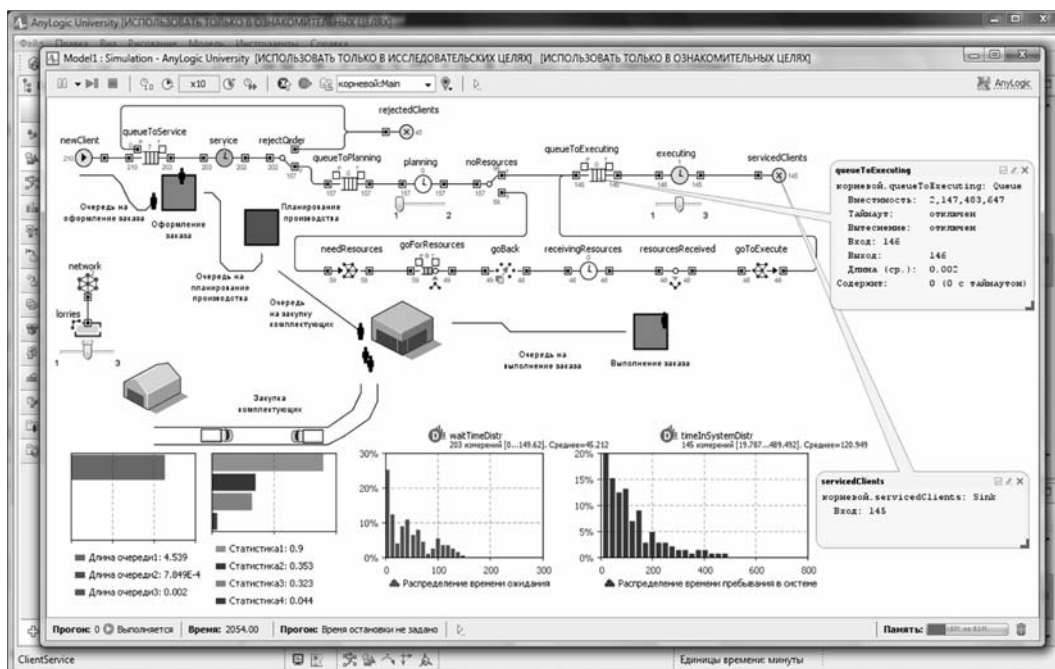


Рис. 5. Анимация СИМ-модели в среде AnyLogic

достаточно, заявка оказывается в очереди на выполнение заказа (*queueToExecuting*), после чего начинается само выполнение (СЧВ — длительность выполнения заказа). По завершении выполнения заказа обслуженные заявки накапливаются в блоке *serviceClients*. Если же ресурсов недостаточно, в блоке *receivingResources* моделируется их закупка (длительность и сумма закупки комплектующих СЧВ), после чего заявка поступает в очередь *queueToExecuting*.

Блоки *network*, *lorries*, *needResources*, *goForResources*, *goBack*, *resourcesReceived* и *goToExecute* используются для моделирования транспортной сети, т. е. процессов закупки комплектующих с помощью имеющихся транспортных средств. Во всех необходимых элементах предусмотрены механизмы генерации СЧВ по заданному закону распределения (можно задать тип закона и его параметры).

В СИМ-модели для оптимизации выделены следующие параметры: число специалистов отдела планирования; число специалистов и число оборудования производственного отдела; число транспортных средств. Под соответствующими блоками расположены "бегунки", с помощью которых можно в реальном времени изменять значения параметров от минимального до максимального. AnyLogic имеет широкие возможности для визуализации СИМ-моделей: в ПО присутствует достаточно большая библиотека изображений, кроме того, любые изображения можно нарисовать самому ЛПР, причем анимация возможна и в режиме 3D.

Анимацию построенной СИМ-модели иллюстрирует центральная часть рис. 5. С помощью ломаных линий здесь отображены четыре очереди: на

оформление заказа, на планирование производства, на закупку комплектующих и на выполнение заказа. Заявки показаны как черные человеческие фигурки; квадраты — это блоки обслуживания заявок: отдел србыта, плановый отдел, производственный отдел. Когда отдел простаивает, квадрат на рис. 5 имеет зеленый цвет, во время обслуживания — красный цвет. Наиболее важный отдел закупок изображен в более подробном виде: как грузовики, дорога, зоны погрузки и разгрузки. При работе СИМ-модели, когда возникает необходимость закупки комплектующих и при наличии свободных транспортных средств, одно из них отправляется за комплектующими — в левый нижний бокс на рис. 5. Если все грузовики заняты, заявка ожидает освобождения одного из них возле бокса в центре. "Играя" значениями трех параметров, можно наблюдать, как это влияет на очереди и скорость обслуживания — чтобы таким образом подбирать наилучшие (квази-оптимальные) параметры бизнес-процесса.

Нажав мышью на любой элемент СИМ-модели, можно вывести окно с параметрами и статистикой по нему (см. правую часть рис. 5). Также предусмотрена возможность сбора и отображения статистики работы модели в виде графиков и диаграмм (см. нижнюю часть рис. 5). Можно наблюдать, например, за длиной очереди, статистикой занятости любого обслуживающего устройства, распределением времени пребывания заявки в СМО и т. д. Для управления временем моделирования предусмотрена соответствующая панель — после запуска СИМ-модели можно остановить, ускорить или замедлить (для очень медленных или очень быстрых процессов) ее работу в целях более углубленного и

детального изучения объекта в удобном онлайн-режиме [7].

Построение СИМ-модели начинается путем создания ее в меню *Файл* → *Создать* → *Модель*, где задаются имя модели и место ее сохранения. Завершается создание модели нажатием кнопки "Далее" после выбора пункта меню "Начать с нуля". Активные объекты являются основными блоками иерархической модели AnyLogic, которая отображается в виде дерева в панели "Проекты": сама модель образует верхний уровень дерева; эксперименты, классы активных объектов и Java-классы образуют нижележащий уровень; элементы, входящие в состав активных объектов, вложены в соответствующую подветвь дерева класса активного объекта и т. д.

Активный объект *Main* можно раскрыть в режиме редактора, чтобы начать построение модели с переноса из панели *Палитра* элементов библиотек. В основной библиотеке использованы следующие элементы:

- *Source* — создающий заявки;
- *Queue* — моделирующий очередь заявок;
- *Recourse Pool* — задающий набор ресурсов, которые могут захватываться и освобождаться заявками;
- *Service* — который захватывает для заявки заданное количество ресурсов, задерживает заявку, а затем освобождает захваченные ею ресурсы;
- *Delay* — задерживающий заявки на заданный промежуток времени;
- *Select Output* — направляющий входящие заявки в один из двух выходных портов в зависимости от выполнения заданного условия;
- *Select Output 5* — объект, который направляет входящие заявки в один из пяти выходных портов в зависимости от выполнения заданных (детерминистических или соответствующих заданным вероятностям) условий.

Каждому элементу из библиотеки необходимо задать его свойства в одноименном окне. Имеется возможность изменить название элемента, при необходимости выбрать его отображение на рабочем поле, задать закон распределения, фигуру анимации и т. д. Затем элементы соединяются между собой и проводится тестирование модели. Информация о выявленных ошибках содержится в окнах *Консоль* и *Ошибки*. После устранения ошибок и выполнения пробных прогонов ЛПР может приступить к последующим этапам СИМ.

Измерить время, проведенное заявкой между двумя точками схемы на рис. 5, можно с помощью элементов *Time Measure Start* и *Time Measure End*, которые необходимо установить после элемента *Source* и перед элементом *Sink*, чтобы собрать статистику времени пребывания заявки в процессе. Результаты СИМ из панели *Палитра* библиотеки *Статистика* выводятся на рабочее поле в виде гистограммы, которая в нашем случае (см. нижнюю

zakaz filiala : Optimization

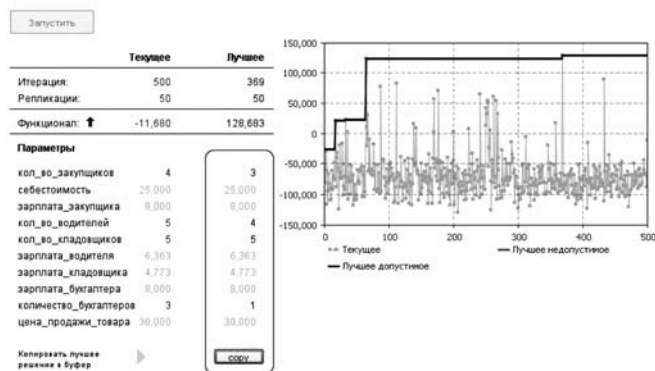


Рис. 6. К реализации оптимизационного СИМ-эксперимента в среде AnyLogic

часть рис. 5) показывает распределение времени обработки заявок. Далее в модель добавляются необходимые параметры и переменные: в данном случае параметрами являются число закупщиков, зарплата и себестоимость, как расчетные переменные заданы чистая прибыль и число выполненных заявок. В настройках также задается модельное время, в течение которого будет наблюдаться поведение СИМ-модели [7].

Возможны два вида СИМ-эксперимента: простой и оптимизационный. В простом эксперименте задаются априори выбранные значения параметров, целью оптимизационного эксперимента является подбор наилучшего значения для выбранного параметра [2, 8]. В нашем случае модельное время составляет 10 080 мин (семь календарных дней), поскольку выполнение отдельных заявок может достигать нескольких суток. В результате работы СИМ-модели определяется оптимальное число сотрудников — закупщиков, кладовщиков, водителей, бухгалтеров для каждого филиала рассматриваемой торговой компании.

При простом эксперименте число сотрудников сразу появляется в окне с результатами СИМ. Целевой функцией при оптимизационном СИМ-эксперименте (рис. 6) является чистая прибыль филиала компании, которую нужно максимизировать. Для параметра "число закупщиков" имеется ограничение: от 1 до 7 чел., в очереди на обслуживание не должно скапливаться больше 10 заявок. По итогам СИМ оптимальное число закупщиков — 3, водителей — 4, кладовщиков — 5, бухгалтеров — 1.

Заключение

Степень соответствия требованиям и возможности рассмотренных средств ПО позволяют использовать их для реализации СИМ-моделей по МДМ в целях повышения эффективности работы СУ современными сложными системами организационно-технического типа [1, 2]. Для проведе-

ния СИМ в интересах управления бизнес-процессами выделены универсальный язык программирования Delphi и среда имитационного моделирования AnyLogic.

Высокая гибкость и скорость работы при относительно невысокой стоимости обеспечивает Delphi конкурентные преимущества перед другими языками программирования несмотря на сложность его изучения и отсутствие возможности анимации. Среду AnyLogic выгодно отличают наличие большого числа встроенных объектов и широкие возможности анимации моделируемых бизнес-процессов при приемлемых показателях скорости и гибкости, главным ее недостатком остается высокая стоимость. Представлены примеры использования данных средств ПО при разработке СИМ-моделей объектов в области телекоммуникаций, добычи природных энергоносителей и оптово-розничной торговли.

Авторы выражают признательность канд. техн. наук, доц. Р. Р. Халимову и С. В. Луковкину за участие в совместных исследованиях. Настоящая статья, к глубокому сожалению его постоянных соавторов, стала последней прижизненной работой канд. техн. наук, доц. Ю. В. Трошина.

Список литературы

1. Димов Э. М., Маслов О. Н., Трошин Ю. В. Снижение неопределенности выбора управленческих решений с помощью метода статистического имитационного моделирования // Информационные технологии. 2014. № 6. С. 51–57.
2. Димов Э. М., Маслов О. Н., Пчеляков С. Н., Скворцов А. Б. Новые информационные технологии: подготовка кадров и обучение персонала. Ч. 2. Имитационное моделирование и управление бизнес-процессами в инфокоммуникациях. Самара: Изд-во СНЦ РАН, 2008. 350 с.
3. Лоу А. М., Кельтон В. Д. Имитационное моделирование / Пер. с англ. СПб.: Питер; Киев: BHV, 2004. 847 с.
4. Кониух В. Л., Зиновьев В. В., Игнатьев Я. Б. Методы имитационного моделирования дискретных систем. Обзор программных продуктов. Кемерово: Изд. КемНЦ СО РАН. 2003. URL: <http://www.gpss.ru/immod05/sl/konyuh/print.html> (дата обращения 20.09. 2014).
5. Димов Э. М., Луковкин С. В., Халимов Р. Р. Анализ средств имитационного моделирования бизнес-процессов // Телекоммуникации. 2010. № 8. С. 43–48.
6. Емельянов А. А., Власова Е. А., Дума Р. В. Имитационное моделирование экономических процессов. М.: Финансы и статистика, 2002. 368 с.
7. Димов Э. М., Маслов О. Н., Трошин Ю. В., Халимов Р. Р. Динамика разработки имитационной модели бизнес-процесса // Инфокоммуникационные технологии. 2013. Т. 11, № 1. С. 63–78.
8. Ануфриев Д. П., Димов Э. М., Маслов О. Н., Халимов Р. Р. Сравнительная эффективность методов и средств информационной поддержки управленческих решений // Инфокоммуникационные технологии. 2014. Т. 12, № 1. С. 54–67.

Ad. M. Dimov, Professor, O. N. Maslov, Head of Chair, e-mail: maslov@psati.ru,

Yu. V. Troshin, Associate Professor

Povolzhskiy State University of Telecommunications and Informatics

Statistical Simulation Modeling Process Software Choice

The article considers the problem of statistical simulation modeling (SSM) software choice for complex organizational-technical systems. The examples of provided SSM-models applying universal language Delphi and simulation environment AnyLogic are presented. The SSM business process model of paying customers service of Telecommunications Company is the first example of a programming language Delphi. The second example — is SSM business process model of resuscitating (repairing) of oil wells. The "Order Fulfillment" business process of trading company modeled in the AnyLogic. Presented SSM models are destined for practical application by managers of telecommunication, oil-producing and trading business.

Keywords: statistical simulation modeling method, software, universal language Delphi, simulation environment AnyLogic

References

1. Dimov E. M., Maslov O. N., Troshin Yu. V. Snizhenie neopredelennosti vybora upravlencheskikh reshenij s pomoshh'yu metoda statisticheskogo imitatsionnogo modelirovaniya. *Informatsionnye tekhnologii*. 2014. N. 6. P. 51–57.
2. Dimov E. M., Maslov O. N., Pchelyakov S. N., Skvortsov A. B. *Novye informatsionnye tekhnologii: podgotovka kadrov i obuchenie personala*. Ch. 2. Imitatsionnoe modelirovanie i upravlenie biznes-protsessami v infokommunikatsiyakh. Samara: Izd-vo SNTS RAN, 2008. 350 p.
3. Lou A. M., Kel'ton V. D. *Imitatsionnoe modelirovanie*: Per. s angl. SPb.: Piter; Kiev: BHV, 2004. 847 p.
4. Konyukh V. L., Zinov'ev V. V., Ignat'ev YA. B. *Metody imitatsionnogo modelirovaniya diskretnykh sistem. Obzor programnykh produktov*. Кемерово: Изд. КемНЦ СО РАН. 2003. URL: [http://](http://www.gpss.ru/immod05/sl/konyuh/print.html)

www.gpss.ru/immod05/sl/konyuh/print.html (data obrashheniya 20.09. 2014).

5. Dimov E. M., Lukovkin S. V., Khalimov R. R. Analiz sredstv imitatsionnogo modelirovaniya biznes-protsessov. *Telekommunikatsii*. 2010. N. 8. P. 43–48.
6. Emel'yanov A. A., Vlasova E. A., Duma R. V. *Imitatsionnoe modelirovanie ekonomicheskikh protsessov*. М.: Финансы и статистика, 2002. 368 p.
7. Dimov E. M., Maslov O. N., Troshin Yu. V., Khalimov R. R. Dinamika razrabotki imitatsionnoj modeli biznes-protsessa. *Infokommunikatsionnye tekhnologii*. 2013. V. 11, N. 1. P. 63–78.
8. Anufriev D. P., Dimov E. M., Maslov O. N., Khalimov R. R. Sravnitel'naya ehffektivnost' metodov i sredstv informatsionnoj podderzhki upravlencheskikh reshenij. *Infokommunikatsionnye tekhnologii*. 2014. V. 12, N. 1. P. 54–67.