

© Воронкин Р.А., Скоробогатов С.А.
Voronkin R., Skorobogatov S.

ПРИМЕНЕНИЕ ОБЪЕКТНО-РЕЛЯЦИОННОГО СООТВЕТСТВИЯ В ЗАДАЧАХ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ СИСТЕМ МАССОВОГО ОБСЛУЖИВАНИЯ С НАЛИЧИЕМ САМОПОДОБИЯ

USING OBJECT-RELATIONAL IMPEDANCE MISMATCH IN THE SYSTEMS OF MASS SERVICE WITH PRESENCE OF SELFRESEMBLANCE SIMULATION MODELING PROBLEM

Аннотация. В статье описаны объектно-реляционного соответствия бизнес-объектов и таблиц базы данных (БД), программы имитационного моделирования системы массового обслуживания. В качестве такой технологии используется LINQ to SQL, позволяющая инициировать работу бизнес-объектов имитационного моделирования с поддержкой их перманентности.

Annotation. In article description object-relational impedance mismatch business-object and database tables (BD) for the system of mass service simulation modeling program. As such technologies is used LINQ to SQL, allowing initiate functioning a business-object simulation modeling with support their persistent.

Ключевые слова. Моделирование систем массового обслуживания, объектно-реляционное соответствие, LINQ to SQL.

Key words. System of mass service modeling, object-relational impedance mismatch, LINQ to SQL.

Введение

Экспериментальные исследования и анализ многочисленных измерений информационных потоков на пакетном уровне указывают на специфическую природу процессов в сетях связи, не укладывающуюся в традиционные рамки известных случайных моделей. Этим обстоятельством объясняются многочисленные исследования, проводимые в настоящее время с целью изучения влияния различных факторов, приводящих к возникновению сложных сетевых процессов.

Характерным для описания процессов передачи данных пакетным трафиком являются обнаруженные на практике свойства самоподобия или масштабной инвариантности статистических характеристик.

В связи с обнаружением этих особенностей сетевых процессов особую актуальность приобретают вопросы разработки конструктивных методов исследования самоподобия применительно к современным сетям связи и учет влияния на характер формирования управ-

ляющих воздействий при передаче пакетного трафика [1].

В этих условиях разработка новых сетевых технологий и повышение эффективности работы современных телекоммуникационных систем требуют создания имитационных моделей, наиболее полно отражающих отмеченные выше свойства сетевых процессов.

Цель статьи. Разработка программы, моделирующей работу многоканальной системы массового обслуживания с ограниченной очередью при обслуживании самоподобного трафика методом статистических испытаний с возможностью хранения результатов экспериментов в базе данных для повторного их использования.

Изучение предметной области. Предметная область – некоторая часть элементов реального мира, данные о которых используются информационной системой.

Каждый из объектов предметной области обладает определенным набором свойств (атрибутов), которые описываются при составлении инфологической модели. Для построения инфологической модели разрабо-

Воронкин Роман Александрович – кандидат технических наук, доцент кафедры защиты информации Северо-кавказского государственного технического университета, тел. 8-928-300-13-31;

Скоробогатов Сергей Александрович – преподаватель Ставропольского военного института связи ракетных войск, тел. 8-905-490-11-32.

Voronkin Roman – the candidate of the technical sciences, the assistant professor of the department of Data Protection of the North-Caucasian State Technical University, tel. 8-928-300-13-31;

Skorobogatov Sergey Aleksandrovich - the teacher of the Stavropol Military Institute of Communication of Missile Troops, tel. 8-905-490-11-32.

танной программы требовались основные контролируемые элементы: сущность, атрибут, связь. Множество сущностей предметной области разбивается на группы объектов, однородных по структуре и поведению и называемых типами сущностей.

Так как одной из задач при моделировании сетей связи методом статистических испытаний является имитация входного потока заявок, потока времени обслуживания заявок, задаваемых генераторами случайных чисел согласно условиям задачи, то необходимо выделить сущность «Генератор случайных чисел» (таблица Ran-domGenerators), в состав которой входят следующие атрибуты:

- код генератора (поле GeneratorID);
- сигнатура генератора (поле GeneratorSign);
- параметр 1 генератора (поле Param1);
- параметр 2 генератора (поле Param2);
- параметр 3 генератора (поле Param3).

Для хранения информации о созданных в результате моделирования экспериментах с возможностью их повторного использования была выделена сущность «Эксперимент» (таблица Experiments), включающая атрибуты:

- код эксперимента (поле ExperimentID);
- количество каналов (поле ChanelAmount);
- размер накопителя (поле DriveSize);
- квант времени (поле TimeQuantum);
- код генератора случайных чисел, определяющий интервал времени поступления заявок в систему (поле SourceGeneratorID);
- код генератора случайных чисел, определяющий интервал времени обслуживания заявок в канале системы (поле ChanelGeneratorID);
- количество генерируемых заявок (поле DemandsNumber);
- комментарий, упрощающий поиск экспериментов (поле Remarks).

Для хранения результатов моделирования была выделена сущность «Результат имитационного моделирования» (таблица Results), включающая следующие атрибуты:

- код результата (поле ResultID);
- код эксперимента, для которого получен результат (поле ExperimentID);
- дата и время окончания имитационного моделирования (поле ResultTime);
- вероятность отказа заявки (поле RefusalProbability);
- среднее число занятых каналов (поле

OccupiedChanelNumber);

- среднее число заявок в накопителе (поле DemandsAmountInDrive);
- среднее время пребывания заявки в накопителе (поле DemandsTimeInDrive);
- среднее число заявок в системе (поле DemandsAmount in System);
- среднее время пребывания заявки в системе (поле DemandsTimeInSystem).

Ввиду того, что функционирование статистической модели СМО основано на рассмотрении состояния системы через фиксированные промежутки времени t_i , $i= 0, 1, 2, \dots, N$, была выделена еще одна сущность – временная метка (таблица Times tamps), состоящая из следующих атрибутов:

- код временной (поле Times tamp ID);
- код результата имитационного моделирования (поле Result ID);
- время, соответствующее временной метке в относительных единицах (поле Time);
- общее число сгенерированных источником заявок до указанного момента времени (поле TotalDemands);
- общее число успешно обслуженных заявок до указанного момента времени (поле SuccessDemands);
- общее число заявок, получивших отказ до указанного момента времени (поле FailDemands);
- число заявок в накопителе в момент времени, соответствующий временной метке (поле Demands in Drive);
- число обслуживаемых в каналах системы заявок в момент времени, соответствующий временной метке (поле Demands in Chanel).

Даталогическая модель базы данных, представляющая ее в виде множества взаимосвязанных отношений, приведена на рис. 1. В этой модели поддерживаются связи между отношениями «один ко многим» и «один к одному». Причем последние описывают отношения агрегации сущностей «Эксперимент» и «Генератор случайных чисел», показывая таким образом, что для получения результатов имитационного моделирования требуется один генератор, описывающий свойства поступающего потока заявок, и второй – для определения длительности обслуживания заявок каждым каналом. С каждым экземпляром сущности «Эксперимент» связано несколько экземпляров сущности «Результат имитационного моделирования», тогда как один экземпляр последней связан с несколькими экземплярами сущности «Временная метка», содержащей характеристики системы массового

обслуживания в определенный момент времени, из промежутка, в течение которого проводится моделирование. Совокупность временных меток позволяет проанализировать динамику работы системы и выявить ее особенности. Тем не менее, поскольку временные метки хранятся в базе данных (БД), то динамика системы в таком описании обладает свойством перманентности.

В качестве языка разработки выбран язык программирования C#. C# – язык программирования, сочетающий объектно-ориентированные и аспектно-ориентированные концепции [2]. Компилятор с C# входит в стандартную установку самой .NET, поэтому программы на нём можно создавать и компилировать даже без инструментальных средств вроде Visual Studio.

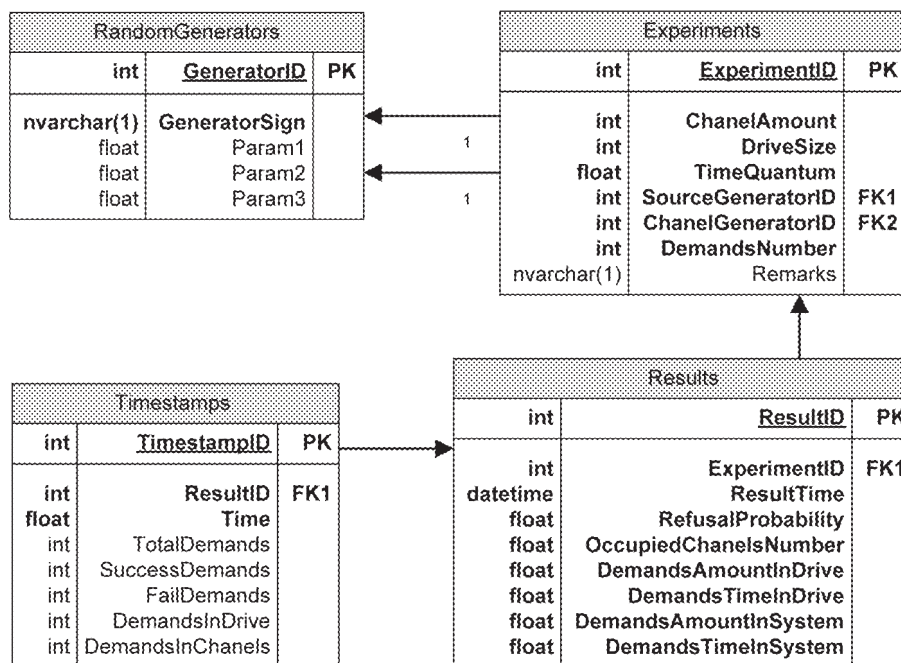


Рис. 1. Дatalogическая модель базы данных

Выбор СУБД. При выборе системы управления базами данных (СУБД) были проанализированы преимущества и недостатки многих современных средств управления базами данных. В итоге выбор был остановлен на СУБД Microsoft SQL Server Compact Edition 2008.

Microsoft SQL Server Compact Edition 2008 – система управления реляционными базами данных, разработанная корпорацией Microsoft. СУБД не является «клиент-серверной» и предназначена для взаимодействия с малыми БД. Запускается внутри процесса программы.

Выбор среды разработки проекта. Для разработки программы, моделирующей работу многоканальной СМО была выбрана среда разработки Microsoft Visual Studio .NET 2008 и язык программирования C#. Visual Studio представляет собой профессиональный инструмент высокого уровня. В нем сочетается простота версий Express и мощные средства разработки, необходимые для создания клиентских приложений, работающих с данными, многоуровневых клиент-серверных приложений с использованием веб-служб, и разнообразных веб-приложений.

Для нормального функционирования создаваемого программного продукта необходимо наличие следующих установленных программ:

- ОС Windows XP/7/Vista;
- NET Framework 3.5;
- Microsoft SQL Server Compact Edition 2008.

Обращение к данным, хранящимся в БД, производится в программе с помощью технологии LINQ to SQL [3], которая служит для выполнения объектно-реляционного соответствия (ORM – object – relational impedance mismatch) информации из БД с бизнес-объектами .NET. Большинство инструментов ORM пытаются абстрагировать физическую БД в виде бизнес-объектов. С такой абстракцией обычно теряется возможность выполнения запросов SQL, составляющих значительную часть абстракции реляционных БД. Именно это и отличает LINQ to SQL от большинства его аналогов. Преимущество заключается не только в получении бизнес – объектов, которые отображаются на БД, но и также в полноценном языке запросов, подобном хорошо известному SQL. Схема ORM для разработанной программы показана на рис. 2.

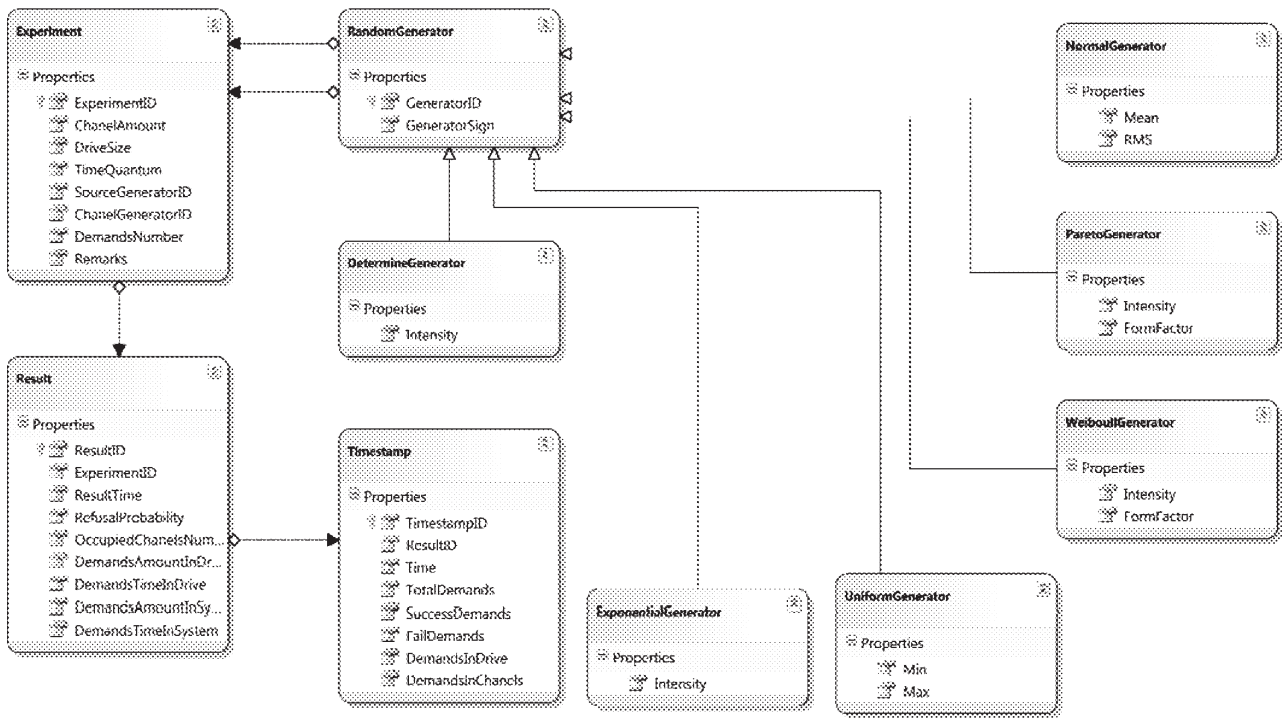


Рис. 2. Схема DBML объектно-реляционного соответствия LINQ to SQL таблиц базы данных и бизнес-объектов программы

Выводы

Таким образом, была спроектирована информационная база данных программной системы, выполняющей имитационное моделирование сетевых процессов. Дальнейшее выполнение объектно-реляционного соответ-

ствия между бизнес-объектами программы и таблицами БД с помощью технологии LINQ to SQL позволяет инкапсулировать работу бизнес-объектов имитационного моделирования с поддержкой их перманентности.

Литература

1. Воронкин Р.А., Скоробогатов С.А. Применение программных средств в задачах моделирования многоканальных систем массового обслуживания с неограниченной очередью, обслуживающих самоподобную нагрузку. Сборник научных трудов СевКавГТУ. Серия «Естественно-научная». № 6 // Северо-Кавказский государственный технический университет, Под ред. Синельникова Б.Н.- Ставрополь: СевКавГТУ, 2010. – с. 24-30
2. Patten – мл., Джозеф С. LINQ: язык интегрированных запросов в C# для профессионалов: Пер. с англ. – М.: ООО «И. Д. Вильямс», 2008. - 560 с.
3. Jenning, Roger. Professional ADO .NET 3.5 with LINQ and Entity Framework. Wiley Publishing, Inc. Indianapolis, Indiana, Canada, 2009. - 676 с.

Материал поступил в редакцию 30. 06. 2010 г.