

ОБРАБОТКА ДАННЫХ И УПРАВЛЕНИЕ В СЛОЖНЫХ СИСТЕМАХ



ВЫПУСК 5

Филиал Кемеровского государственного университета
в г. Анжеро-Судженске

ОБРАБОТКА ДАННЫХ
И УПРАВЛЕНИЕ
В СЛОЖНЫХ СИСТЕМАХ

Выпуск 5

Под редакцией профессора А.Ф. Терпугова

Издательство Томского университета
2003

УДК 519.2

ББК 22.17

О 24

Обработка данных и управление в сложных системах:

О 24 Сборник статей / Под ред. А.Ф. Терпугова. – Томск: Изд-во Том. ун-та, 2003. Вып. 5. – 238 с.

ISBN 5-7511-1646-x

Сборник содержит статьи сотрудников и аспирантов факультета математики и информатики филиала Кемеровского университета в г. Анжеро-Судженске, посвященные статистической обработке временных рядов, актуарной математике, а также вопросам управления в системах массового обслуживания и в измерительных системах.

Для студентов, аспирантов, научных работников, занимающихся вопросами анализа временных рядов и управления в измерительных системах.

УДК 519.2

ББК 22.17

ISBN 5-7511-1646-x

© Филиал Кемеровского государственного
университета в г. Анжеро-Судженске, 2003

СОДЕРЖАНИЕ

Войтиков К.Ю., Змеев О.А., Моисеев А.Н. Основные функциональные требования к подсистеме «Брокер объектных запросов» в рамках унифицированного процесса разработки программного обеспечения ©	3
Гарайшина И.Р., Назаров А.А. Исследование переходных режимов в математической модели изменения числа лиц, застрахованных в Пенсионном фонде ©.	14
Егоров М.Н., Якупов Р.Т. Прямой подход к решению задачи оптимальной фильтрации состояний нескольких дискретных линейных динамических систем при наблюдении разностей их векторов состояния ©	21
Злобина С.Л. Динамическая модель формирования заданных структурных характеристик социальной системы ©	32
Змеев О.А. Математическая модель деятельности фонда социального страхования с детерминированными расходами на социальные программы при релейно-гистерезисном управлении капиталом ©	42
Колоусов Д.В. Исследование потока заявок, отправленных в источник повторных вызовов сети связи случайного доступа с конечным числом станций © ..	57
Кощоруба П.И. Исследование времени и распределения вероятностей стабильного функционирования неустойчивой сети связи ©	67
Кузнецов Д.Ю. Математическая модель двухлинейной сети связи с вероятностным распределением запросов, управляемая адаптивным протоколом случайного множественного доступа ©	78
Лях Е.А., Назаров А.А., Туренова Е.Л. Численное исследование сети связи с протоколом случайного множественного доступа ©	86
Марголис Н.Ю. Исследование нестабильных сетей случайного доступа ©	91
Масяйкин С.А. Построение переговорного множества при конкурентном взаимодействии страховых компаний и обратной зависимости интенсивности потока рисков от страховых взносов ©	96
Моисеев А.Н., Хохлов А.С. Бизнес-модель документооборота Администрации города ©	105
Назаров А.А. Период занятости в бесконечнолинейных немарковских системах обслуживания ©	119
Назаров А.А., Цой С.А. Исследование математической модели двухканальной сети случайного доступа ©	124
Никитина М.А. Исследование немарковской модели сети связи, управляемой h_2 -настойчивым протоколом случайного доступа с дискретным контролем оповещения о конфликте ©	136
Новицкая Е.В. Организация анализа оборотных средств в системе «1С. Предприятие. Оперативный учет» ©	144
Орлов А.Б. Расчет характеристик бесконечнолинейной системы массового обслуживания с дважды стохастическим входящим потоком ©	150
Орлов А.Б. Средняя длительность периода занятости системы массового обслуживания с вытеснением заявки при дважды стохастическом входящем потоке © ..	158

Поддубный В.В. Максимально правдоподобное полиномиальное сглаживание рядов эмпирических частот ©	171
Терпугов А.Ф., Щирова Н.П. Релейно-гистерезисное управление ценой товара при наличии склада (диффузионное приближение) ©	183
Токарева Е.Г. Двумерное распределение Коши ©	191
Токарева Е.Г. Асимптотические свойства статистик для проверки гипотезы о равенстве дисперсий двух гауссовских процессов ©	195
Чернядьева Н.В. Существование и признаки оптимальности траекторий участников рынка труда ©	202
Яковлева Н.М. Об одной модели равновесия на рынке труда, учитывающей рабочее и досуговое время ©	212
Янковский Б.Е. Энтропийные числовые характеристики случайных величин © ..	222
Глухова Е.В. Вероятностные характеристики количества вредных веществ в организме ©	226

ОСНОВНЫЕ ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ К ПОДСИСТЕМЕ «БРОКЕР ОБЪЕКТНЫХ ЗАПРОСОВ» В РАМКАХ УНИФИЦИРОВАННОГО ПРОЦЕССА РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

К.Ю. ВОЙТИКОВ, О.А. ЗМЕЕВ, А.Н. МОИСЕЕВ

Архитектура большинства современных информационных систем корпоративного уровня построена с применением шаблона Layers [1]. Этот шаблон позволяет организовывать крупномасштабные структурные элементы системы в отдельные уровни со взаимосвязанными обязанностями таким образом, чтобы на более низком уровне располагались низкоуровневые службы и службы общего назначения, а на более высоких уровнях – объекты уровня бизнес-логики приложения. Под уровнем понимается крупномасштабный элемент системы, который очень часто состоит из нескольких пакетов или подсистем. При такой организации информационной системы взаимодействие между уровнями происходит сверху вниз, связывание объектов в обратном направлении является нежелательным.

В [2, 3] предпринята попытка классификации наиболее общих уровней корпоративной информационной системы. В этих работах предполагается выделение в проекте таких подсистем, как уровень интерфейса, уровень управления данными и уровень бизнес-логики. В данной работе в рамках Унифицированного процесса разработки программного обеспечения [4] определяются основные функциональные требования к одному из пакетов обобщенной архитектуры – уровню хранения данных.

Рабочий процесс определения требований является одним из ключевых моментов разработки программного обеспечения. Настоящая работа придерживается методики проведения этого процесса, изложенной в [5]. В качестве результата

рабочего процесса необходимо получить набор вариантов использования и актеров, полностью описывающий основные функциональные возможности разрабатываемой подсистемы.

Анализ проблемы

Выделение уровня хранения в отдельную подсистему связано не только с требованиями шаблона Layers, но и с таким немаловажным достоинством компонентной разработки, как повторное использование. Рассмотрим существующие в настоящее время механизмы и технологии хранения объектов.

Объектные базы данных. С одной стороны, при хранении информации в объектной базе данных не требуется никаких дополнительных служб обеспечения взаимодействия с базой данных. Это составляет ключевое преимущество данного подхода. Но с другой стороны, в настоящий момент объектные базы данных практически недоступны для рядового разработчика.

Реляционные базы данных. Традиционный подход, который применяется для хранения данных. При этом возникает множество проблем, связанных с несоответствием объектно-ориентированного представления данных в системе и их представления в виде набора записей в реляционных таблицах базы данных. Для работы с такими базами данных требуются специальные средства преобразования форм представления информации.

Другие базы данных. Иногда приходится хранить информацию в другом виде, в качестве примера можно рассмотреть очень популярную на сегодняшний день тему XML баз данных. В этом случае возникают аналогичные проблемы с преобразованием форм представления информации, что и при работе с реляционными базами данных, так как при этом возникает несоответствие между представлением постоянно хранимого объекта в системе и его представлением в неobjектно-ориентированной структуре. И в этом случае необходимо создание специальной службы.

Одним из методов решения вышеперечисленных проблем является создание некоторой универсальной службы для реализации и управления подсистемы хранения данных. Для этих целей необходимо разработать контур интерфейса с базой данных или брокер объектных запросов. Брокер объектных запросов – это многократно используемый и обычно расширяемый набор классов, обеспечивающий обслуживание постоянно хранимых объектов.

Существуют известные универсальные брокеры объектных запросов, например основанные на CORBA. Спецификация CORBA предназначена для поддержки различных механизмов реализации объектов, поэтому структура ядра брокера не определяется. Вместо этого задается набор интерфейсных функций, которые должны присутствовать в каждой реализации БОЗ и тем самым маскируют отличия между различными реализациями брокеров объектных запросов. Однако он не поддерживает основного механизма материализации и дематериализации объектов системы.

В настоящей работе рассмотрены основные функциональные требования к брокеру объектных запросов, задачей которого является стандартизация хранения объектов в реляционной базе данных.

В рамках настоящей работы программная подсистема «Брокер» выполняет преобразование объектной формы представления информации в форму записей или другой структурированный формат данных, типа XML, и их сохранение в базе данных, а также некоторые другие операции. Обязательное ограничение, накладываемое на эту подсистему: структура базы данных и способы работы с ней не изменяются. Основное назначение брокера объектных запросов – это материализация и дематериализация объектов системы.

Выявление требований

Один из методов выявления требований к информационной системе заключается в следующем: собираются все заинтересованные стороны (заказчики, исполнители) и формулируют требования к разрабатываемой системе в виде одного –

двух предложений. Затем полученные требования классифицируются заинтересованными сторонами по принципу: обязательно, желательно, не обязательно. После этого команда разработчиков «проводит черту», оставляя из всех получившихся требований только половину наиболее значимых. Затем получившиеся требования ранжируются командой разработчиков по срокам и рискам [5]. То есть требования с наибольшим риском и временем реализации ставятся на «первое место» и реализуются в дальнейшем в первую очередь. Таким образом, получается план работ на первую итерацию. В нашем случае результат этой работы приведен в таблице.

Формировать пакет данных для передачи информации
Расшифровать пакет принятых данных
Позволять добавлять новые объекты
Позволять добавлять в систему с помощью открытого интерфейса новые классы
Обеспечить администратору возможность удаления данных
Обеспечить уникальность идентификаторов объектов в системе
Поддерживать механизм обмена COM
Позволять изменять данные объектов
Позволять помечать данные на удаление
Поддерживать структуру базы данных, соответствующую ОО-подходу к проектированию
Просмотр баз данных
Обеспечить уникальность идентификаторов данных объектов
Обеспечивать независимость структуры БД от разрабатываемых приложений
Предоставлять механизм настройки доступа и запросов к новым форматам хранения данных
Поддерживать пакеты данных в формате OleVariant
Обеспечить подключение к SQL серверу
Проверять уровень и возможность доступа к данным в соответствии с Пб
Поддерживать пакеты данных в формате XML

Поддерживать визуальное проектирование БД в нотации UML
Предоставить администратору механизмы настройки политики безопасности
Использовать хранимые процедуры при работе с SQL сервером
Обеспечивать блокировку данных на уровне записи при редактировании
Поддерживать работу с данными, хранящимися в формате XML
Просмотр текущего состояния сервера (подключение пользователя, открытые объекты и т.п.)
Обеспечить подключение к локальным БД
Обрабатывать запросы клиентов, созданные на специальном языке запросов
Динамически генерировать методы обработки данных в случае работы с локальными БД
Поддерживать работу с распределенной БД
Поддерживать кэширование
Реализовывать механизмы транзакций для локальных БД
Поддерживать специализированный язык запросов клиентов
Поддерживать визуальное проектирование БД в нотации ERD
Поддерживать визуальное проектирование БД в нотации ldf1.x
Управление текущим состоянием (отключение пользователей, блокировка объектов, остановка сервера сообщения пользователям)
Просмотр журналов работы сервера
Поддерживать механизм обмена COBRA
Обеспечивать архивацию данных
Поддерживать автономную работу
Поддерживать атрибуты временной актуальности
Вести журнал транзакций при работе с локальными БД
Вести журнал регистрации пользователей при работе с локальными БД. Вести учет отработанного времени пользователями системы

Таким образом, для первой версии брокера объектных запросов были оставлены требования, выделенные в таблице, а именно: разрабатываемая подсистема должна предоставлять интерфейс другим приложениям для сохранения данных объектов, предоставлять открытый интерфейс для подключения базы данных с известной структурой.

Для разработчиков систем, использующих данный «Брокер», необходимо предусмотреть возможность проектирования схем данных. Для этого следует создать средства проектирования (например, визуального в нотации UML).

Так как основное назначение брокера объектных запросов – это материализация и дематериализация объектов системы, то необходимо, чтобы система обеспечила возможность добавления в систему новых объектов, изменение данных существующего объекта, удаление объектов. Кроме того, брокер объектных запросов должен осуществлять выборку данных описанных в нем объектов. Так как будем предполагать его использование в приложениях, построенных на архитектуре клиент-сервер, то желательным является обеспечение блокировок и кэширования данных. Согласно полученным требованиям необходимо, чтобы «Брокер» обеспечил подключение не только к локальным базам, но и к базам SQL сервера, а также поддерживал механизмы удаленного вызова DCOM.

Реализация функциональных требований в виде вариантов использования

Последним этапом технологического процесса определения функциональных требований является их преобразование в модель прецедентов. Ниже приведены диаграммы вариантов использования для подсистемы «Брокер» с необходимыми комментариями.

Для администрирования построенных с использованием брокера систем необходимо обеспечить администратору возможность определения классов, просмотра базы данных, настройки политики безопасности. Обобщим выделенные требования в виде диаграммы вариантов использования (рис. 1).

Уточнение этого варианта использования изображено на рис 2.

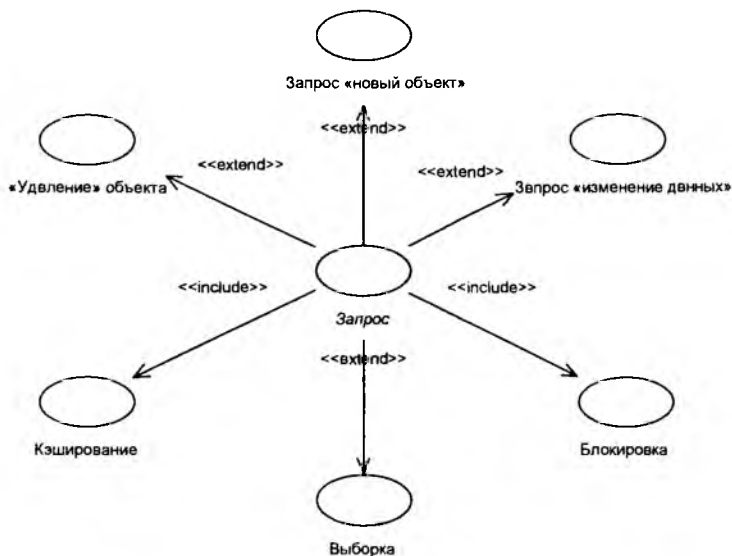


Рис. 2

Согласно рис. 1 в разрабатываемой системе существуют функции, отвечающие за работу с данными, и функции, отвечающие за настройку системы. Таким образом, можно попытаться разделить функции разрабатываемого «Брокера» на отдельные программные модули, что приведет к первоначальному определению архитектуры подсистемы.

Некоторые моменты архитектуры

Одним из модулей системы будет «Сервер». В его обязанности войдет работа с данными. На рис. 3 выделены прецеденты, относящиеся к данному модулю.

Вкратце охарактеризуем модуль «Сервер». Он является не визуальным программным модулем, инициализируется только один раз на компьютере-сервере и закрывается после того, как последний пользователь покинет систему. Это осуще-

ствляется путем поддержки удаленного вызова DCOM. К одной из основных задач данного модуля можно отнести поддержку пакетов обмена данными с приложениями.

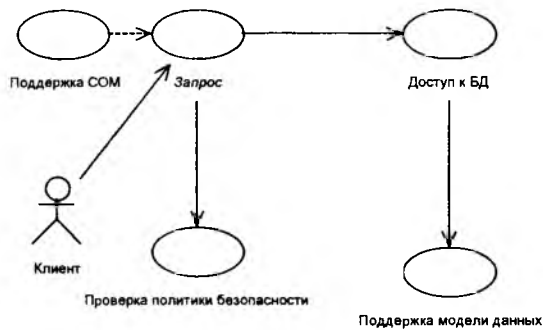


Рис. 3

Как и любой программный модуль, «Брокер объектных запросов» нуждается в первоначальной настройке, такой как определение классов системы, настройка на формат данных, удаление данных, настройка политики безопасности. Все эти вышеизложенные функции можно выделить в программный модуль «Администрирование». На рис. 4 представлены варианты использования, относящиеся к данному программному модулю.

Этот программный модуль используется администратором не только для первоначальной настройки системы, но и для осуществления некого контроля, например просмотра данных и мониторинга.

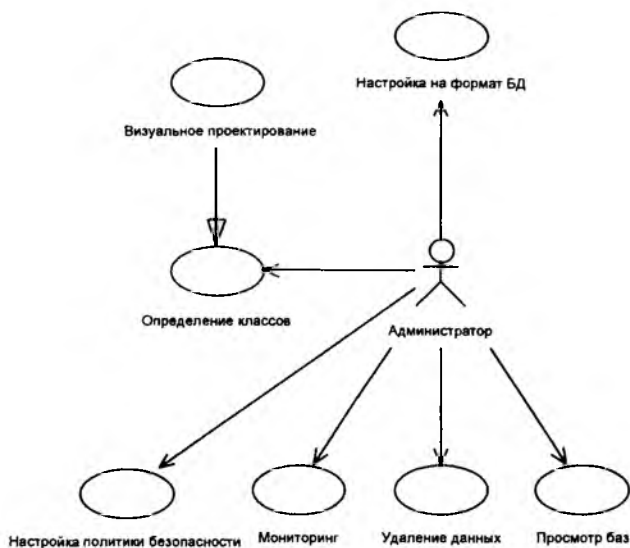


Рис. 4

Таким образом, подведем итоги и сделаем вывод о том, что создаваемый «Брокер объектных запросов» будет состоять из двух подсистем:

1. «Сервер» – к его основным функциям можно отнести: поддержку удаленных вызовов DCOM, обеспечение независимости структуры базы данных от разрабатываемых приложений, поддержку пакетов обмена с приложениями.

2. «Конфигуратор» – к его основным функциям можно отнести: работу с классами (определение новых, изменение существующих), обеспечение подключения к хранилищам данных, а также обеспечение администратору возможности просмотра статуса работающего приложения, оповещений пользователей, принудительные блокировки.

Результатом работ в фазе анализа на первой итерации является документ «Видение». В этом документе отражены функциональные требования к разрабаты-

ваемой системе, ее возможности и назначение. Далее предполагается перейти к следующей фазе.

ЛИТЕРАТУРА

1. Ларман К. Примененне UML и шаблонов проектирования. М.: Издательский дом «Вильямс», 2001. 496 с.
2. Коуд П., Норт Д., Мейфилд М. Объектные модели. Стратегии, шаблоны и приложения. М.: Лори, 1999. 434 с.
3. Змеев О.А., Моисеев А.Н. Шаблоны диаграммы компонентов информационной системы корпоративного уровня. // Вестник ТГУ. Томск, 2002.
4. Якобсон А., Буч Г., Рамбо Дж. Унифицированный процесс разработки программного обеспечения. СПб.: Питер, 2002. 496 с.
5. Леффингуэлл Д., Уидриг Д. Принципы работы с требованиями к программному обеспечению. Унифицированный подход. М.: Издательский дом «Вильямс», 2002. 448 с.