
Шаблоны интеграции корпоративных приложений

*Проектирование, создание
и развертывание решений, основанных
на обмене сообщениями*

Enterprise Integration Patterns

*Designing, Building,
and Deploying Messaging
Solutions*

Gregor Hohpe
Bobby Wolf

With Contributions by
Kyle Brown
Conrad F. D’Cruz
Martin Fowler
Sean Neville
Michael J. Rettig
Jonathan Simon



ADDISON-WESLEY

*Boston • San Francisco • New York • Toronto • Montreal
London • Munich • Paris • Madrid • Capetown • Sydney
Tokyo • Singapore • Mexico City*

Шаблоны интеграции корпоративных приложений

*Проектирование, создание
и развертывание решений, основанных
на обмене сообщениями*

Грегор Хоп
Бобби Вульф

при участии
Кайла Брауна
Конрада Ф. Д'Круза
Мартина Фаулера
Шона Невилла
Майкла Дж. Реттига
Джонатана Саймона



Москва • Санкт-Петербург • Киев
2007

ББК 32.973.26-018.2.75

X78

УДК 681.3.07

Издательский дом “Вильямс”

Главный редактор *С.Н. Тригуб*

Зав. редакцией *В.Р. Гинзбург*

Перевод с английского и редакция *А.В. Журавлева, Н.Н. Селиной*

По общим вопросам обращайтесь в Издательский дом “Вильямс” по адресу:

info@williamspublishing.com, <http://www.williamspublishing.com>

115419, Москва, а/я 783; 03150, Киев, а/я 152

Хоп, Грегор, Вульф, Бобби.

X78 Шаблоны интеграции корпоративных приложений. : Пер. с англ. — М. : ООО “И.Д. Вильямс”, 2007. — 672 с. : ил. — Парал. тит. англ.

ISBN 978-5-8459-1146-9 (рус.)

В данной книге исследуются стратегии интеграции корпоративных приложений с помощью механизмов обмена сообщениями. Авторы рассматривают шаблоны проектирования и приводят практические примеры интеграции приложений, демонстрирующие преимущества обмена сообщениями и эффективность решений, создаваемых на основе этой технологии. Каждый шаблон сопровождается описанием некоторой задачи проектирования, обсуждением исходных условий и представлением элегантного, сбалансированного решения. Авторы подчеркивают как преимущества, так и недостатки обмена сообщениями, а также дают практические советы по написанию кода подключения приложения к системе обмена сообщениями, маршрутизации сообщений и мониторинга состояния системы.

Книга ориентирована на разработчиков программного обеспечения и системных интеграторов, использующих различные технологии и продукты для обмена сообщениями, такие как Java Message Service (JMS), Microsoft Message Queuing (MSMQ), IBM WebSphere MQ, Microsoft BizTalk, TIBCO, WebMethods, SeeBeyond, Vitria и др.

ББК 32.973.26-018.2.75

Все названия программных продуктов являются зарегистрированными торговыми марками соответствующих фирм.

Никакая часть настоящего издания ни в каких целях не может быть воспроизведена в какой бы то ни было форме и какими бы то ни было средствами, будь то электронные или механические, включая фотокопирование и запись на магнитный носитель, если на это нет письменного разрешения издательства Addison-Wesley.

Authorized translation from the English language edition published by Addison-Wesley, Copyright © 2004.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from the publisher.

Russian language edition is published by Williams Publishing House according to the Agreement with R&I Enterprises International, Copyright © 2007.

ISBN 978-5-8459-1146-9 (рус.)

ISBN 0-321-20068-3 (англ.)

© Издательский дом “Вильямс”, 2007

© Pearson Education, Inc., 2004

Оглавление

Введение	24
Глава 1. Решение задач интеграции с помощью шаблонов проектирования	43
Глава 2. Стили интеграции	77
Глава 3. Системы обмена сообщениями	91
Глава 4. Каналы обмена сообщениями	127
Глава 5. Построение сообщений	167
Глава 6. Практикум: простой пример обмена сообщениями	205
Глава 7. Маршрутизация сообщений	243
Глава 8. Преобразование сообщений	339
Глава 9. Практикум: сложный обмен сообщениями	373
Глава 10. Конечные точки обмена сообщениями	477
Глава 11. Управление системой	549
Глава 12. Практикум: управление системой	583
Глава 13. Шаблоны интеграции на практике	609
Глава 14. Кое-что в заключение	629
Приложение А. Список шаблонов проектирования	653
Приложение Б. Шаблоны интеграции корпоративных приложений	660
Основные источники информации	662
Предметный указатель	668

Содержание

Введение	24
Что такое обмен сообщениями	25
Что такое система обмена сообщениями	26
Преимущества обмена сообщениями	27
Недостатки асинхронного обмена сообщениями	30
Мыслим асинхронно	31
Распределенное приложение или интеграция приложений	32
Коммерческие системы обмена сообщениями	32
Форма шаблонов	34
Диаграммы, использованные в книге	36
Примеры и практикумы	38
Как организована эта книга	39
С чего начать	40
Поддержка	41
Резюме	41
Глава 1. Решение задач интеграции с помощью шаблонов проектирования	43
Необходимость интеграции	43
Трудности интеграции	44
Роль интеграционных шаблонов проектирования	46
Типы интеграционных задач	46
Слабое связывание	50
Пример простой интеграции	51
Компоненты слабосвязанного интеграционного решения	55
Пример: “Приборы и устройства”	56
Внутренние системы	57
Размещение заказов	58
Обработка заказов	59
Проверка состояния заказа	65
Изменение адреса клиента	69
Обновление каталога товаров	71
Рассылка новостей	72
Тестирование и мониторинг	73
Резюме	75
Глава 2. Стили интеграции	77
Введение	77
Критерии интеграции приложений	77
Способы интеграции приложений	79
Передача файла (File Transfer)	80
Общая база данных (Shared Database)	83
Удаленный вызов процедуры (Remote Procedure Invocation)	85
Обмен сообщениями (Messaging)	87

Глава 3. Системы обмена сообщениями	91
Введение	91
Основные концепции обмена сообщениями	91
Об организации книги	92
Канал сообщений (Message Channel)	93
Сообщение (Message)	98
Каналы и фильтры (Pipes and Filters)	102
Конвейерная обработка	104
Параллельная обработка	104
История архитектуры каналов и фильтров	105
Маршрутизатор сообщений (Message Router)	109
Типы маршрутизаторов сообщений	111
Транслятор сообщений (Message Translator)	115
Уровни преобразования	116
Уровни связывания	118
Цепочечные преобразования	118
Конечная точка сообщения (Message Endpoint)	124
Глава 4. Каналы обмена сообщениями	127
Введение	127
Основные вопросы, связанные с применением каналов обмена сообщениями	127
Решения	128
Канал “точка-точка” (Point-to-Point Channel)	131
Канал “публикация-подписка” (Publish-Subscribe Channel)	134
Канал типа данных (Datatype Channel)	139
Канал недопустимых сообщений (Invalid Message Channel)	143
Канал недоставленных сообщений (Dead Letter Channel)	147
Гарантированная доставка (Guaranteed Delivery)	149
Адаптер канала (Channel Adapter)	154
Мост обмена сообщениями (Messaging Bridge)	159
Шина сообщений (Message Bus)	162
Глава 5. Построение сообщений	167
Введение	167
Сообщение с командой (Command Message)	169
Сообщение с данными документа (Document Message)	171
Сообщение о событии (Event Message)	174
Запрос-ответ (Request-Reply)	177
Обратный адрес (Return Address)	182
Идентификатор корреляции (Correlation Identifier)	186
Цепочка сообщений (Message Sequence)	192
Срок действия сообщения (Message Expiration)	198
Индикатор формата (Format Indicator)	201
Глава 6. Практикум: простой пример обмена сообщениями	205
Введение	205
Запрос-ответ	205

8 Содержание

Публикация–подписка	206
Запрос–ответ (JMS)	208
Описание	208
Код	210
Обработка сообщения недопустимого формата	216
Итог	217
Запрос–ответ (.NET)	219
Описание	219
Код	221
Обработка сообщения недопустимого формата	225
Итог	226
Публикация–подписка (JMS)	227
Наблюдатель (Observer)	227
Наблюдатель (Observer) в распределенной среде	228
Публикация–подписка	229
Сравнение двух подходов	232
Модель с активным и пассивным источниками данных	233
Проектирование каналов	238
Итог	241
Глава 7. Маршрутизация сообщений	243
Введение	243
Простые маршрутизаторы	243
Составные маршрутизаторы	245
Архитектурные шаблоны	245
Выбор маршрутизатора сообщений	246
Маршрутизатор на основе содержимого (Content-Based Router)	247
Уменьшение зависимостей	249
Фильтр сообщений (Message Filter)	253
Фильтры с сохранением или без сохранения состояния	254
Функции фильтрации, встроенные в системы обмена сообщениями	255
Использование фильтров сообщений вместо маршрутизатора на основе содержимого	256
Динамический маршрутизатор (Dynamic Router)	259
Список получателей (Recipient List)	264
Надежность	267
Динамический список получателей	268
Эффективность с точки зрения сети	269
Список получателей или канал “публикация–подписка” с фильтрами сообщений	269
Разветвитель (Splitter)	274
Итеративные разветвители	275
Статические разветвители	276
Упорядоченные или неупорядоченные дочерние сообщения	277
Агрегатор (Aggregator)	283
Детали реализации	285
Стратегии агрегации	286
Преобразователь порядка (Resequencer)	297
Порядковый номер	298

Внутреннее устройство	299
Борьба с переполнением буфера	300
Обработчик составного сообщения (Composed Message Processor)	307
Рассылка-сборка (Scatter-Gather)	310
Карта маршрутизации (Routing Slip)	314
Применение карты маршрутизации к унаследованным приложениям	319
Область применения	320
Реализация простого маршрутизатора с картой маршрутизации	320
Диспетчер процессов (Process Manager)	325
Управление состоянием	327
Экземпляры процесса	327
Корреляция	328
Сохранение состояния в сообщениях	329
Создание определения процесса	330
Сравнение диспетчера процессов с другими шаблонами	332
Брокер сообщений (Message Broker)	334
Глава 8. Преобразование сообщений	339
Введение	339
Устранение зависимостей	340
Управление метаданными	340
Преобразование данных в других схемах интеграции	341
Оболочка конверта (Envelope Wrapper)	342
Расширитель содержимого (Content Enricher)	348
Фильтр содержимого (Content Filter)	354
Квитанция (Claim Check)	358
Выбор ключа	360
Использование квитанции для сокрытия информации	361
Использование квитанции с диспетчером процессов	362
Нормализатор (Normalizer)	364
Определение формата сообщения	365
Каноническая модель данных (Canonical Data Model)	367
Способы преобразования	369
Двойное преобразование	370
Разработка канонической модели данных	370
Зависимости между форматами данных	371
Глава 9. Практикум: сложный обмен сообщениями	373
Пример: кредитный брокер	373
Выбор процентной ставки по кредиту	373
Проектирование потока сообщений	375
Режим выполнения операций: синхронный или асинхронный	377
Адресация: распределение или аукцион	378
Стратегии агрегации: несколько каналов или один	380
Управление одновременным доступом	380
Три реализации	381
Синхронная реализация с использованием Web-служб	383
Архитектура решения	383
Соображения по поводу проектирования Web-служб	384

Набор средств Apache Axis	387
Поиск расположения службы	390
Приложение кредитного брокера	391
Компоненты приложения кредитного брокера	393
Клиентское приложение	409
Запуск решения	410
Ограничения производительности	412
Ограничения данного примера	412
Резюме	413
Асинхронная реализация с использованием MSMQ	414
Экосистема локального брокера	414
Закладываем основы: шлюз обмена сообщениями	415
Базовые классы для общей функциональности	418
Компонент банка	423
Компонент кредитного бюро	425
Компонент кредитного брокера	426
Оптимизация кода кредитного брокера	445
Запуск системы	448
Повышение производительности	449
Несколько слов о тестировании	453
Ограничения данного примера	456
Резюме	457
Асинхронная реализация с использованием TIBCO ActiveEnterprise	458
Архитектура решения	458
Набор средств, используемых для реализации	460
Интерфейсы	464
Реализация синхронных служб	465
Процесс кредитного брокера	468
Управление параллельными аукционами	472
Выполнение	473
Резюме	475
Глава 10. Конечные точки обмена сообщениями	477
Введение	477
Шаблоны отправки и получения сообщений	477
Шаблоны потребителей сообщений	478
Некоторые аспекты реализации конечных точек	480
Шлюз обмена сообщениями (Messaging Gateway)	482
Соединение шлюзов в цепочки	486
Обработка исключений системы обмена сообщениями	486
Генерация кода шлюзов	487
Использование шлюзов в процессе тестирования	487
Преобразователь обмена сообщениями (Messaging Mapper)	491
Упрощение кодирования	495
Преобразователь или транслятор	495
Транзакционный клиент (Transactional Client)	498
Опрашивающий потребитель (Polling Consumer)	507
Событийно управляемый потребитель (Event-Driven Consumer)	511
Конкурирующие потребители (Competing Consumers)	515

Диспетчер сообщений (Message Dispatcher)	521
Избирательный потребитель (Selective Consumer)	528
Постоянный подписчик (Durable Subscriber)	535
Идемпотентный получатель (Idempotent Receiver)	541
Активатор службы (Service Activator)	545
Глава 11. Управление системой	549
Введение	549
Мониторинг и управление	550
Наблюдение и анализ трафика сообщений	550
Тестирование и отладка	551
Шина управления (Control Bus)	552
Обходной путь (Detour)	556
Отвод (Wire Tap)	558
Журнал доставки сообщения (Message History)	561
Хранилище сообщений (Message Store)	565
Интеллектуальный заместитель (Smart Proxy)	567
Тестовое сообщение (Test Message)	577
Вентиль канала (Channel Purger)	579
Глава 12. Практикум: управление системой	583
Управление системой кредитного брокера	583
Оснащение кредитного брокера средствами контроля	583
Управляющая консоль	584
Качество обслуживания кредитного брокера	585
Проверка функциональности кредитного бюро	592
Перемещение при сбое кредитного бюро	598
Усовершенствование управляющей консоли	600
Ограничения примера, приведенного в этой главе	607
Глава 13. Шаблоны интеграции на практике	609
Практикум: система формирования цен на облигации	609
Построение системы	610
Архитектура с учетом шаблонов	610
Структура каналов	616
Выбор канала сообщений	620
Решение проблемы с помощью шаблонов	623
Мигающие ячейки	623
Сбой системы в процессе работы	625
Резюме	628
Глава 14. Кое-что в заключение	629
Новые стандарты и перспективы интеграции корпоративных приложений	629
Связь между стандартами и шаблонами проектирования	630
Основные организации, занимающиеся утверждением стандартов	631
Компоненты бизнес-процессов и внутренний обмен сообщениями между Web-службами	633
eXML и ebMS	635
BPEL4WS	638

12 Содержание

WSCI	640
Стандарты компонентов бизнес-процессов для Java	641
Спецификации WS-*	643
Заключение	651
Приложение А. Список шаблонов проектирования	653
Приложение Б. Шаблоны интеграции корпоративных приложений	660
Основные источники информации	662
Предметный указатель	668

*Моей семье и всем моим друзьям, которые поддерживали
меня во время написания этой книги.*

—Грегор

Шэрон, моей любящей жене.

—Бобби

Предисловие

Джона Крупи

Как вы реагируете на появление новой технологии? Вы изучаете ее. Столкнувшись с платформой J2EE, я не нашел ничего лучшего, кроме как обратиться к ее спецификациям (книг по J2EE на то время еще не существовало). В частности, я сконцентрировал свое внимание на подмножестве J2EE — технологии EJB. Тем не менее изучение технологии является лишь первым шагом к ее эффективному применению. Преимущество платформенных технологий состоит в том, что они ограничивают разработчика определенным кругом задач. К сожалению, это не исключает неверного применения самой технологии.

По моим наблюдениям, за последние 15 лет в умах разработчиков программного обеспечения прочно укоренились идеи *эффективного* программирования и проектирования. В настоящее время на рынке существует огромное количество книг по программированию на Java и C#, однако гораздо меньше по проектированию приложений. Написав книгу *Core J2EE Patterns*, мы с Дипаком Алуrom (Deepak Alur) и Дэном Малксом (Dan Malks) хотели помочь J2EE-разработчикам в создании более эффективного кода. Основная идея нашей книги заключалась в использовании шаблонов проектирования. Как отметил Джеймс Бейти (James Baty), главный инженер компании Sun Microsystems, “шаблоны являются краеугольным камнем проектирования”. Я полностью согласен с этим высказыванием. К счастью, так же считают и авторы настоящей книги.

Книга, которую вы держите в руках, посвящена чрезвычайно популярной в последнее время теме: интеграции приложений с помощью обмена сообщениями. На мой взгляд, идея обмена сообщениями найдет свое применение не только при интеграции приложений, но и при создании Web-служб. Как это было на заре появления технологий J2EE и .NET, вопросам проектирования Web-служб уделяется пока еще очень мало внимания. Многие считают, что Web-службы являются всего лишь новым способом решения существующих задач интеграции, и я полностью с этим согласен. Тем не менее это не отменяет необходимость проектирования самих Web-служб. Изюминка данной книги состоит в наличии большого количества шаблонов, которые можно применять при проектировании Web-служб и других интеграционных систем. Бобби и Грегор не балуют читателя конкретными примерами спецификаций Web-служб, поскольку они все еще нуждаются в доработке и утверждении. Впрочем, это вполне простительно. Подлинная ценность книги проявится после объявления соответствующих спецификаций стандартами, что позволит использовать предложенные шаблоны проектирования для разработки соответствующих этим стандартам приложений. Вполне вероятно, что создание архитектур, ориентированных на использование служб, станет основным способом интеграции приложений в ближайшем будущем.

Прочтите эту книгу от начала до конца. Она способна кардинально изменить вашу карьеру.

Джон Крупи
Бетесда, Мэриленд
Август 2003 г.

Предисловие

Мартина Фаулера

В процессе написания книги *Patterns of Enterprise Application Architecture* (Фаулер М. *Архитектура корпоративных программных приложений*. — М. : Издательский дом “Вильямс”, 2004) мне посчастливилось получить подробную рецензию на включенный в нее материал от Кайла Брауна (Kyle Brown) и Рейчел Рейниц (Rachel Reinitz). В результате нескольких неофициальных встреч в офисе Кайла в Роли-Дарем мы пришли к выводу, что самым существенным пробелом моей книги стали системы асинхронного обмена сообщениями.

Безусловно, моя книга содержит множество пробелов и не претендует на звание исчерпывающего источника информации по шаблонам проектирования корпоративных приложений. Тем не менее отсутствие в ней описания систем асинхронного обмена сообщениями особенно болезненно ввиду того, что эти системы будут играть ключевую роль при решении задач интеграции. Приложения не могут существовать обособленно одно от другого, и нам нужны методики, позволяющие наладить связь между приложениями, изначально не предусматривавшими возможность взаимодействия.

На сегодняшний день существует множество различных технологий интеграции приложений. На мой взгляд, наиболее многообещающей из них является обмен сообщениями. Организация эффективного обмена сообщениями — нетривиальная задача, сложность которой обусловливается асинхронной природой сообщений, а также различиями в подходах к асинхронному программированию.

К сожалению, мне не хватило времени, сил и, что самое главное, знаний, чтобы достойно раскрыть тему обмена сообщениями в *Patterns of Enterprise Application Architecture*. Тем не менее я все-таки нашел решение этой проблемы в виде Грегора и Бобби, взявшихся за написание книги, которую вы сейчас держите в руках.

Следует отметить, что авторы блестяще справились с поставленной перед ними задачей. Если вы уже работали с системами обмена сообщениями, настоящая книга поможет вам систематизировать накопленный опыт и знания. Если же вы только приступаете к изучению данной темы, книга поспособствует закладке прочного фундамента, который пригодится вам при использовании любой технологии обмена сообщениями.

Мартин Фаулер
Мелроуз, Массачусетс
Август 2003 г.

Вступление

Эта книга посвящена интеграции корпоративных приложений с помощью обмена сообщениями. В ней не рассматриваются конкретные технологии или продукты. Вместо этого она ориентирована на разработчиков и интеграторов, использующих различные технологии и продукты для обмена сообщениями, такие как:

- MOM- и EAI-решения IBM WebSphere MQ, Microsoft BizTalk, TIBCO, WebMethods, SeeBeyond, Vitria и др.;
- реализации спецификации Java Message Service (JMS), внедренные в серверы приложений J2EE, а также в отдельные продукты;
- служба Microsoft Message Queuing (MSMQ), доступная через различные API, например через классы пространства имен Microsoft .NET System.Messaging;
- новые стандарты Web-служб, поддерживающие асинхронный обмен сообщениями (например, WS-ReliableMessaging), а также соответствующие API, такие как Java API for XML Messaging (JAXM) от Sun Microsystems или Web Services Extensions (WSE) от Microsoft.

Интеграция корпоративных приложений означает нечто большее, чем создание распределенного приложения с *n*-уровневой архитектурой. Тогда как отдельный уровень распределенного приложения не может выполняться самостоятельно, интегрированные приложения представляют собой независимые сущности, способные к взаимной координации по принципу слабой связи. Обмен сообщениями позволяет нескольким приложениям передавать данные и команды по сети в соответствии с подходом “отправил и забыл”. Последнее позволяет вызывающему приложению отправить информацию и вернуться к выполнению текущей задачи, переложив всю ответственность за доставку информации на систему обмена сообщениями. При необходимости вызывающее приложение может быть уведомлено о результате доставки информации с помощью функции обратного вызова. Несмотря на то что по сравнению с синхронным обменом сообщениями асинхронный подход способен усложнить проектирование приложения, возможность неограниченного числа повторов асинхронного вызова способна существенно повысить надежность коммуникаций. Асинхронный обмен сообщениями обладает и другими преимуществами, такими как регулирование числа запросов и балансировка нагрузки.

Для кого предназначена эта книга

Эта книга призвана помочь разработчикам приложений и системным интеграторам наладить взаимодействие между приложениями с помощью средств, ориентированных на обмен сообщениями. В первую очередь, она предназначена для следующих специалистов.

- **Архитекторы и разработчики, проектирующие и создающие сложные корпоративные приложения**, которые нуждаются в интеграции с другими приложениями. Мы ориентируемся на разработчиков приложений, использующих современные платформы, такие как Java 2 Enterprise Edition (J2EE) и Microsoft .NET. С помощью этой книги вы научитесь соединять приложение с уровнем обмена сообщениями для взаимодействия с другими приложениями. Следует отметить, что предметом рассмотрения этой книги является исключительно интеграция приложений, а не их создание. Шаблоны проектирования, применяющиеся при создании корпоративных приложений, рассматриваются в книге Мартина Фаулера *Patterns of Enterprise Application Architecture* (Фаулер М. *Архитектура корпоративных программных приложений*. — М. : Издательский дом “Вильямс”, 2004).
- **Архитекторы и разработчики, проектирующие и создающие интеграционные решения**, предназначенные для налаживания взаимодействия между пользовательскими приложениями. Вероятно, большинство читателей этой книги уже сталкивались с коммерческими средствами интеграции приложений, такими как IBM WebSphere MQ, TIBCO, WebMethods, SeeBeyond, Vitria и др. Каждое из этих средств содержит реализацию многих шаблонов проектирования, представленных в книге. Помимо фундаментальных концепций, стоящих за интеграцией приложений, принимать правильные решения при проектировании вам поможет универсальный язык шаблонов, не зависящих от конкретного вендора.
- **Корпоративные архитекторы**, в обязанности которых входит мониторинг программных и аппаратных активов предприятия. Язык и графическое представление шаблонов проектирования помогут описать крупномасштабные решения интеграции, охватывающие множество различных технологий. Кроме того, язык шаблонов поможет наладить эффективное взаимодействие между корпоративным архитектором и архитекторами/разработчиками корпоративных приложений и решений интеграции.

Чему учит эта книга

Основное назначение этой книги — познакомить читателя с принципами проектирования успешных решений интеграции корпоративных приложений. В процессе ее чтения вы узнаете:

- преимущества и ограничения асинхронного обмена сообщениями в сравнении с другими технологиями интеграции;
- как определить каналы обмена сообщениями, в которых нуждается приложение; регламентировать процедуру получения одного и того же сообщения несколькими потребителями; обрабатывать сообщения неправильного формата;

- когда следует отправлять сообщение, какую информацию оно должно содержать и как использовать специальные свойства сообщений;
- как доставить сообщение в конечную точку его назначения (в том числе и при отсутствии информации о получателе сообщения);
- как преобразовать сообщение при несовпадении формата отправителя и получателя;
- как спроектировать код соединения приложения с системой обмена сообщениями;
- как проводить управление и мониторинг системы обмена сообщениями.

Что не рассматривается в этой книге

На наш взгляд, все книги, в названиях которых встречается слово “корпоративный”, делятся на три категории. К первой категории относятся книги, авторы которых стараются всецело охватить предмет обсуждения, однако в конечном итоге останавливаются на подробном рассмотрении конкретных решений. Книги второй категории содержат практические советы по разработке конкретных решений, ограничивая при этом диапазон рассматриваемого материала. Наконец, можно попытаться написать книгу, объединяющую в себе лучшие характеристики первых двух категорий, однако в этом случае она либо не будет закончена, либо будет опубликована настолько поздно, что потеряет свою актуальность. Настоящая книга относится ко второй категории. Мы попытались написать книгу о создании интеграционных решений, пожертвовав при этом некоторыми смежными темами. В частности, мы не рассматривали вопросы безопасности, сложного отображения данных, рабочих потоков, механизма поддержки правил, масштабируемости, устойчивости, а также распределенной обработки транзакций (продукты XA, Tuxedo и др.). Асинхронный обмен сообщениями был выбран основной темой книги по нескольким причинам. Во-первых, это одна из наиболее перспективных технологий интеграции приложений, которой свойственно множество интереснейших вопросов проектирования. Во-вторых, рассмотрение асинхронного обмена сообщениями позволяет абстрагироваться от реализации конкретных решений интеграции, предлагаемых различными вендорами.

Эта книга не является руководством по конкретной технологии обмена сообщениями или межплатформенному решению. Чтобы подчеркнуть многообразие концепций, представленных в книге, мы включили в нее примеры использования целого ряда различных технологий, таких как JMS, MSMQ, TIBCO, BizTalk и XSL. Каждая из перечисленных технологий рассматривалась исключительно в аспекте принятия решений проектирования и возникающих при этом компромиссов. Для подробного изучения какой-либо из технологий интеграции приложений обратитесь к источникам, перечисленным в конце книги, или к ресурсам Интернета.

Структура книги

Как следует из названия книги, большая ее часть посвящена *шаблонам*. Шаблоны — это проверенный способ представления накопленного опыта и знаний в таких областях, как архитектура программных приложений, объектно-ориентированное проектирование и создание интеграционных решений с помощью технологий обмена сообщениями.

Каждый шаблон состоит из описания некоторой проблемы проектирования, обсуждения исходных условий и представления элегантного, сбалансированного решения. В большинстве случаев предлагаемое решение является результатом длительного процесса поиска. Каждый шаблон “вбирает” в себя опыт, накопленный старшими разработчиками и архитекторами в попытке дать оптимальный ответ на конкретную задачу интеграции. Хотим подчеркнуть, что в этой книге нет “придуманных” нами шаблонов. Шаблон нельзя “придумать”; к нему можно прийти в результате длительных “полевых” испытаний, извлекая уроки из собственных ошибок.

Если вы уже имеете опыт работы со средствами интеграции корпоративных приложений и с архитектурой асинхронного обмена сообщениями, то большинство из представленных в этой книге шаблонов наверняка покажутся вам до боли знакомыми. Однако даже в этом случае она может принести немалую пользу. Чтение книги поможет закрепить знания, накопленные в результате использования технологий обмена сообщениями, а также поспособствует развитию привычки документирования создаваемых решений и взаимоотношений между ними. Благодаря универсальному языку шаблонов вы сможете наладить эффективное взаимодействие со своими коллегами.

Представленные в книге шаблоны можно реализовать с помощью различных платформ и языков программирования. Чтобы облегчить процесс встраивания шаблона в конкретную среду, в книгу были включены примеры реализации шаблонов с привлечением нескольких популярных технологий, таких как JMS, MSMQ, TIBCO, BizTalk, XSL и др. Кроме того, мы продемонстрировали примеры создания целостных решений, основанных на сочетании шаблонов проектирования.

Интеграция корпоративных приложений с помощью архитектуры асинхронного обмена сообщениями — сложная и увлекательная задача. Искренне надеемся, что наша книга поможет вам блестяще решить ее.

О фотографии на обложке книги

На обложках всех книг серии *A Martin Fowler Signature Book* размещается фотография какого-либо моста. В некотором смысле нам повезло, так как мост великолепно иллюстрирует идею интеграции. На протяжении тысячелетий мосты помогали человеку преодолевать препятствия в виде рек, горных ущелий и оживленных автострад.

Мост Тайко-баси (“Барабанный мост”) перекинут через пруд перед храмом Сумиёситайся, расположенным в Осаке, Япония. Мы выбрали этот мост за его красоту и элегантность. Синтоистский храм Сумиёситайся был построен на берегу бухты Нанива в III веке н.э. и посвящен божествам, которые оберегали моряков, отправлявшихся в плавание. За много лет вода отступила, и сейчас храм удален от воды более чем на 5 км. В начале каждого года Сумиёситайся посещают более трех миллионов человек.

Грегор Хоп
Сан-Франциско, Калифорния

Бобби Вульф
Роли, Северная Каролина

Сентябрь 2003 г.
www.enterpriseintegrationpatterns.com

Ждем ваших отзывов!

Вы, читатель этой книги, и есть главный ее критик. Мы ценим ваше мнение и хотим знать, что было сделано нами правильно, что можно было сделать лучше и что еще вы хотели бы увидеть изданным нами. Нам интересны любые ваши замечания в наш адрес.

Мы ждем ваших комментариев и надеемся на них. Вы можете прислать нам бумажное или электронное письмо либо просто посетить наш Web-сервер и оставить свои замечания там. Одним словом, любым удобным для вас способом дайте нам знать, нравится ли вам эта книга, а также выскажите свое мнение о том, как сделать наши книги более интересными для вас.

Отправляя письмо или сообщение, не забудьте указать название книги и ее авторов, а также свой обратный адрес. Мы внимательно ознакомимся с вашим мнением и обязательно учтем его при отборе и подготовке к изданию новых книг.

Наши электронные адреса:

E-mail: info@williamspublishing.com

WWW: <http://www.williamspublishing.com>

Наши почтовые адреса:

в России: 115419, Москва, а/я 783

в Украине: 03150, Киев, а/я 152