

Вступление: наведение мостов

На обложке этой книги представлена фотография моста через пролив Орезунд, соединяющего Швецию и Данию. Похоже, что на обложках всех книг по архитектуре программного обеспечения должны быть изображения мостов, но в данной книге для этого имеются дополнительные причины.

Этот мост заменил паром, которым я переправлялся еще ребенком. Мне очень нравится каждый раз ездить по нему, хотя я проделывал это десятки раз.

Между прочим, мой отец был среди тех, кто возводил самые высокие участки этого моста.

Но самое главное, что эта книга во многом сродни наведению мостов — в данном случае между пользователями и разработчиками, конкретной предметной областью и программным обеспечением, логикой и хранением информации, теми, кто организует и ведет базы данных, и теми, кто занимается объектно-ориентированным программированием.

Для того чтобы это вступление не выглядело слишком заумным, я, пожалуй, воздержусь от аналогии с шаблоном моста (Bridge) [GoF Design Patterns].

Назначение книги

Основное назначение этой книги состоит в том, чтобы показать, как строится четкая, но удобная с точки зрения сохраняемости модель предметной области (Domain Model). В ней также показано, как может выглядеть решение сохраняемости для такой модели предметной области и, в частности, как навести мост между такой моделью и базой данных.

Иными словами, приступая к этой книге, я поставил перед собой цель перенести содержание книг *Domain-Driven Design* Эрика Эванса [Evans DDD] и *Архитектура корпоративных программных приложений* Мартина Фаулера [Fowler PoEAA] в определенный контекст.

Проектирование на основе предметной области (ППО — Domain-Driven Design (DDD)) можно воспринимать как нечто абстрактное. Поэтому для пояснения, в частности, сохраняемости требуются вполне конкретные примеры. Мои примеры могут показаться довольно простыми, но они закладывают основание в качестве отправной точки. В этой книге не только поясняется, как пользоваться шаблонами, но и объясняется, как это можно делать, например, в объектно-реляционных преобразователях (O/R Mappers).

Для меня стало вполне очевидно, что разработать архитектуру приложений на все случаи жизни невозможно. В этой связи шаблоны оказались достаточно обобщенными, чтобы неоднократно применять их в разных контекстах.

Основное внимание в книге уделяется не самим шаблонам, а их применению в качестве инструмента и языка для обсуждения различных аспектов проектирования. Но даже те читатели, которые не знакомы с шаблонами, получают в ходе чтения книги некоторое представление о них и, возможно, заинтересуются ими.

Это же относится и к разработке посредством тестирования (РПТ — Test-Driven Development (TDD)). Пока еще не все разработчики заинтересовались этим методом. И особенно это касается сообщества программирующих в .NET, где РПТ, как, впрочем, и шаблоны, считается в лучшем случае удобным, специальным методом или же вообще неизвестна. Читатели этой книги смогут научиться РПТ.

Причины для написания книги

Написание моей первой книги [Nilsson NED] превратилось в трудное занятие в дополнение ко всем моим обычным проектам и обязанностям. В этой связи у меня появилась твердая уверенность, что я вряд ли буду писать еще одну книгу, но прошло время и у меня возникла потребность поделиться тем, что осталось еще невысказанным.

Я изменил свое решение после того, как недавно прочитал две книги. Первой из них была книга *Архитектура корпоративных программных приложений* Мартина Фаулера [Fowler PoEAA]. Эта книга вдохновила меня на очередное опробование модели предметной области после ряда предыдущих неудачных попыток.

Затем я прочитал книгу *Domain-Driven Design* Эрика Эванса [Evans DDD]. Эта книга дала мне представление о том, как мыслить и действовать в процессе разработки, уделяя основное внимание предметной области, и, в какой-то степени, как применять шаблон модели предметной области.

Еще одно влияние на мое решение оказали курсы по шаблонам, которые я вел в течение двух лет. В процессе общения с учащимися и работы с учебным материалом у меня сложилось определенное представление о данном предмете.

Мои взгляды на ППО претерпели изменения в ходе работы над весьма любопытным (и, к счастью, уже завершившимся) открытым проектом под названием Valhalla, который я разрабатывал вместе с Кристофером С্কьолдборгом (Christopher Skjoldborg). (Большую часть работы выполнил Кристофер.)

Таким образом, я посчитал, что нужно написать книгу, которая должна быть в большей степени посвящена практике применения, чем теории, но в то же время под нее следует заложить твердое основание, которым могли бы послужить две упомянутые выше книги. Мне больше по сердцу именно практическое применение, поскольку я считаю себя, прежде всего, разработчиком.

Кому адресована книга

Эта книга адресована широкому кругу читателей. Она окажется полезной тем, у кого имеются определенные знания в следующих областях.

- Объектная ориентация
- .NET или аналогичная платформа
- C# или аналогичный язык программирования
- Реляционные базы данных, например, SQL Server

Тем не менее, интерес и увлеченность читателя способны восполнить недостающие знания и опыт работы в перечисленных выше областях.

Попробую подробнее пояснить, почему эта книга рассчитана на широкий круг читателей. Прежде всего, нужно подумать о том, чтобы разработчики не замыкались в рамках используемых платформ. Тем, кто работает в .NET, книга поможет освоить более фундаментальный подход, чем метод типа “перетаскивай, пока не опустишь” (если можно позволить себе такое поверхностное обобщение). А те, кто работает с Java, должны извлечь для себя пользу из описания и примеров совместного использования ППО и объектно-реляционного преобразования.

На мой взгляд, выбор языка проектирования или платформы имеет все меньшее значение, поэтому было бы не совсем логично ориентировать книгу только на тех, кто работает с .NET или Java. Намного полезнее попытаться выяснить круг читателей, подойдя к этому вопросу с другой меркой. И в этом отношении книга адресована руководителям проектов и разработчикам программного обеспечения и его архитектуры.

Эта книга также может оказаться полезной для читателей как промежуточного, так и продвинутого уровня подготовки, а в какой-то степени и для начинающих.

Как организована книга

Эта книга состоит из следующих четырех частей.

Часть I. Основы

В этой части архитектура приложений и процессы рассматриваются в общих чертах. Основное внимание уделяется моделям предметной области и ППО [Evans DDD]. Кроме того, в ней вводятся понятия шаблонов и РПТ. В эту часть включены следующие главы.

- Глава 1. Переоценка ценностей
В этой главе рассматриваются свойства оцениваемой архитектуры приложений и процесса для получения качественных результатов с точки зрения разработки системы. Тема для обсуждения навеяна экстремальным программированием.

- Глава 2. Переход к шаблонам
В этой главе приведены примеры и рассмотрены шаблоны из разных семейств, включая шаблоны проектирования, архитектурные шаблоны и шаблоны предметной области.
- Глава 3. РПТ и рефакторинг кода
В главе 1 уделяется достаточно внимания РПТ и рефакторингу кода, а в этой главе данные вопросы рассматриваются более углубленно и приводятся довольно обширные примеры и разновидности РПТ.

Часть II. Прикладное ППО

В этой части речь идет о прикладном ППО, а также о подготовке модели предметной области для инфраструктуры и соответствующих правилах.

- Глава 4. Новая используемая по умолчанию архитектура
В этой главе перечисляются требования к примеру приложения, а также создается первая пробная модель в качестве основания для материала последующих глав. При этом используется архитектура на основе применяемой модели.
- Глава 5. Проектирование на основе модели предметной области
Требования, сформулированные в предыдущей главе, используются в этой главе в качестве основания для того, чтобы постепенно, выполняя РПТ, приступить к построению модели предметной области в стиле ППО.
- Глава 6. Подготовка к инфраструктуре
Несмотря на то что рассмотрение различных аспектов инфраструктуры мы пытаемся отложить на как можно более дальний срок, о них все же целесообразно подумать заранее и подготовить модель предметной области к инфраструктуре с учетом ее потребностей. В этой главе подробно рассматриваются достоинства и недостатки моделей предметной области без учета сохраняемости.
- Глава 7. Порядок устанавливающих правила
В этой главе речь идет о бизнес-правилах в форме проверки достоверности, а также об учете в решении на основе модели предметной области потребностей в этих правилах с отсылкой на требования, сформулированные в главе 4.

Часть III. Применение шаблонов архитектуры корпоративных программных приложений

В этой части некоторые шаблоны из книги *Архитектура корпоративных программных приложений* Мартина Фаулера [Fowler PoEAA] переносятся в контекст дискуссии о том, что требуется от инфраструктуры для поддержки сохраняемости в модели предметной области. Выполнение подобных требований рассматривается на примере отдельного инструмента.

- Глава 8. Инфраструктура для сохраняемости
Когда уже имеется довольно удачная модель предметной области, самое время подумать об инфраструктуре, а в данной книге это, главным образом, инфраструктура для сохраняемости. В этой главе рассматриваются различные свойства решений сохраняемости и отнесение определенного решения к конкретной категории.
- Глава 9. Приведение в действие преобразователя NHibernate
Разделение решений на категории, выполненное в предыдущей главе, используется на примере решения сохраняемости, а именно преобразователя NHibernate [NHibernate].

Часть IV. Что дальше

В этой части основное внимание уделяется другим методам проектирования, на которые следует обратить внимание, а затем и применять. Кроме того, здесь показано, как действовать на уровне представления, когда требуется навести мосты между ним и моделью предметной области и как организовать тестирование пользовательского интерфейса на стадии разработки. Эта часть почти полностью написана приглашенными авторами.

- Глава 10. Рекомендуемые методы проектирования
После краткого изложения ограниченного контекста в этой главе рассматриваются методы проектирования, на которые следует обратить внимание, включая ориентацию на обслуживание, внесение зависимостей, инверсию управления, аспектную ориентацию.
- Глава 11. Акцент на пользовательском интерфейсе
В этой главе основное внимание уделяется сопряжению пользовательского интерфейса с моделью предметной области, а также повышению проверяемости пользовательского интерфейса при использовании модели предметной области как в насыщенных клиентских приложениях, так и в Web-приложениях.

Приложения

В двух приложениях к книге представлены дополнительные примеры видов модели предметной области и перечень шаблонов с кратким их описанием.

Основания для примеров на C#

Эту книгу ни в коем случае не следует рассматривать как учебник по C#. Тем не менее этот язык программирования понадобился мне для примеров.

Я выбрал этот язык, главным образом, потому, что пользуюсь в настоящее время именно им, а большинство программирующих на VB.NET и Java могут свободно читать исходный текст и на C#.

Что же касается конкретной версии, то большая часть кода в приведенных примерах вполне работоспособна в версиях C# 1.1 и 2.0, хотя в редких случаях отдельные фрагменты кода относятся к версии 2.0.

Темы, не охваченные в книге

За рамками этой книги осталось немало тем, но, на мой взгляд, в ней не были рассмотрены две важные темы: распределение и усовершенствованное моделирование.

Распределение

Первоначально я планировал включить в книгу материал, посвященный распределению, но затем пришел к выводу, что книга получится слишком распыленной. Тем не менее распределение упоминается в отдельных местах книги.

Усовершенствованное моделирование

Из заглавия книги читатель может сделать вывод, что в ней он найдет интересные примеры усовершенствованного моделирования в некоторых предметных областях. Но это не совсем так. Основное внимание уделяется РПТ и дополнению ППО соответствующей инфраструктурой.

finally{}

Нетрудно догадаться, что такая книга является плодом труда не одного человека, а целого коллектива. В приведенном ниже разделе “Благодарности” перечислены все те, кто принимал участие в работе над книгой и ее выпуске. Тем не менее, все ошибки, которые не удалось обнаружить до сдачи книги в печать, являются моими и только моими, как основного автора.

Дополнительную информацию о книге можно найти по адресу www.jmsk.se/adddp.

Если вернуться к наведению мостов, то фотография моста через пролив Орезунд была сделана моим другом Магнусом фон Шенком (Magnus von Schenk) во время одной из его прогулок по морю под парусом.

Написать эту книгу оказалось легче, чем первую, тем не менее, она стоила мне немало крови, пота и слез. Надеюсь, что она поможет вам сберечь немного того, другого и третьего. Желаю вам получить удовольствие от книги. Удачи!

*Джимми Нильссон
www.jmsk.se/weblog/
Листерби, Швеция
сентябрь 2005 года*

Благодарности

Рискуя забыть кого-нибудь из тех, кого следует непременно здесь упомянуть (в этом случае я заранее приношу свои извинения и благодарю их), я выражаю свою признательность следующим лицам.

- Авторам предисловий: Мартину Фаулеру и Эрику Эвансу.
- Приглашенным авторам: Франсу Боуме, Дану Бистрёму, Уди Дахану (Udi Dahan), Эрику Дёрненбургу (Eric Dürnenburg), Матсу Хеландеру, Ингмару Лундбергу, Филипу Нельсону, Клаудио Перроне (Claudio Perrone), Александару Сеовичу (Aleksandar Seović) и Кристоферу С্কьолдборгу.
- Рецензентам, которые дали мне немало полезных советов: Уильяму Булли (William Bulley), Марку Берхопу (Mark Burhop), Дану Бистрёму, Рассу Кондику (Russ Condick), Энди Конраду (Andy Conrad), Кристиану Кроухэрсту, Майку Дёрфлеру (Mike Dürfler), Стиву Айхерту (Steve Eichert), Эрику Эвансу, Мартину Фаулеру, Полю Гиленсу, Крису Хаддаду (Chris Haddad), Киму Хардингу Кристенсену (Kim Harding Christensen), Матсу Хеландеру, Нираю Гупте (Neeraj Gupta), Андерсу Хесселлунду, Роджеру Йоханссону (Roger Johansson), Роджеру Кратцу (Roger Kratz), Тронду-Эйрику Коллоэну, Ингмару Лундбергу, Патрику Лёвендалю (Patric Løwendahl), Маркусу Макиннесу (Marcus MacInnes), Филипу Нельсону, Перу-Ола Нильссону, Фредрику Нормену (Fredrik Normén), Йохану Нормену (Johan Normén), Майклу О'Брайену (Michael O'Brien), Майклу Платту (Michael Platt), Себастьяну Росу (Sébastien Ros), Мартину Розену-Лидхольму, Александару Сеовичу, Кристоферу С্কьолдборгу, Джорджу Вишу II (George Vish II), Грегори Янгу и Кристену Окессону (Christen Åkesson).
- Художнику-оформителю книги Кьеллу Варнквисту (Kjell Warnquist)
- Редактору Лидии Уэст (Lydia West Zahn)
- Редактору-консультанту Крису Зану (Chris Zahn)
- Редактору проекта Майклу Терстону (Michael Thurston)
- Литературному редактору Элизе Уолтер (Elise Walter)
- Управляющему сбытом Керту Джонсону (Curt Johnson)
- Рецензенту издательства Карен Геттман (Karen Gettman), а также Сондре Скотт (Sondra Scott), которая первой уговорила меня взяться за этот проект.

Без всех этих людей книга не увидела бы свет или, по крайней мере, была бы ужасного качества. Благодарю, друзья. Я перед вами в долгу!

От издательства

Вы, читатель этой книги, и есть главный ее критик и комментатор. 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 интересно услышать и любые другие замечания, которые вам хотелось бы высказать 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 или нет вам эта книга, а также выскажите свое мнение о том, как сделать наши книги более интересными для вас.

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