

**Ю.В. Малинина**

## **ИСПОЛЬЗОВАНИЕ ШАБЛОНОВ ПРИ РАЗРАБОТКЕ WIS<sup>1</sup>**

### **1. ВВЕДЕНИЕ**

В настоящее время WEB является широко распространенным информационным хранилищем. Возрастающая популярность Интернета и быстрое развитие программного обеспечения от любительских Web-сайтов до корпоративных систем заставляют гипермедиа-технику совершенствоваться быстрее, чем когда-либо прежде. Это новое поколение приложений, включающее гипермедиа-функции в сложные системы транзакций, развивается непрерывно; новые версии нуждаются в строгих спецификациях, так как могут повлиять на основу жизнедеятельности компании (как, например, в приложениях e-торговли). Все это приводит к тому, что комплекс гипермедиа-приложений должен разрабатываться с использованием последних достижений технологии программирования. Тем не менее гипермедиа-приложения обладают специфическими характеристиками, отличающими их от стандартного программного обеспечения. Во-первых, поскольку построение их навигационной архитектуры является для них критическим, разработка навигации должна быть выделена как самостоятельная задача и явно специфицирована. Кроме того, гипермедиа-приложения являются диалоговыми по определению и аспекты интерфейса пользователя должны тоже быть описаны при помощи проектных примитивов. Наконец, поскольку рынок навязывает циклы короткой разработки, время становится важным аспектом создания эффективных приложений и, следовательно, многократное использование проектов предпочтительнее, так как позволяет избежать разработки с нуля. Проектный опыт — существенный аспект сокращения времени разработки, и поэтому необходимы средства для представления проектного опыта в виде некоторых формализмов, которые облегчат его повторное использование и распространение среди разработчиков.

Существующая потребность в новых методах, решающих упомянутые выше аспекты в проектировании, не отменяет того, что эти методы должны

---

<sup>1</sup> Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований (грант № 01-01-794) и Министерства образования РФ.

соответствовать существующим стандартам проектирования приложений. К несчастью, состояние искусства проектирования программного обеспечения и некоторых методов, как, например, Unified Modeling Language (UML), терпят неудачу при применении к проектированию гипермедиа-приложений из-за их специфических свойств.

В предлагаемой статье рассматривается использование проектных шаблонов для решения перечисленных проблем. Фактически, целью является рассмотрение процесса применения шаблонов для улучшения проектирования архитектуры Web-информационных систем (WIS).

## 2. ОСОБЕННОСТИ ПРОЕКТИРОВАНИЯ WIS

Проектирование WIS — информационных систем, которые созданы с использованием Web-технологии, — включает множество проблем: определение общей архитектуры приложения, например тип интерфейса (связи, форматы и т.п.) с базами данных или другими системами; организацию информационных узлов таким образом, чтобы пользователь мог легко управлять ими; представление навигации и функционала и т.д.

Тем не менее требуется использовать опробованную технологию программирования для проектирования. Существует много методов, которые могли бы применяться для моделирования некоторых аспектов WIS, как, например, OOHDM [7, 8], RMM [4] и т.п. Они обеспечивают поддержку ссылок и платформенно-независимое проектирование, которые помогают описывать проблемную область так, чтобы гипермедиа-компоненты (узлы, связи и индексы) могли быть затем реализованы при помощи инструментальных средств, запускаемых во время установки. В этих методах есть ясное разделение на методы, представляющие интерес для архитектурного (концептуального) проектирования, методы для проектирования навигации и методы для проектирования интерфейса пользователя. Выделение проектирования навигации в отдельную проблему позволяет разработчикам сконцентрироваться на отличительной черте WIS-гипермедиа-систем.

Однако существуют программные системы, нацеленные на поддержку проектирования WIS, которые игнорируют выделение проектирования навигации как отдельной задачи, обращаясь с ней, как с некоторой функцией интерфейса пользователя. Например, в VisualWave описывается разработка Smalltalk-приложения, где интерфейс содержит элементы управления, допускающие открытие нового окна. Но если воспользоваться этим элементом для операции навигации, то мощность навигации будет потеряна не из-

за характеристики окружения, а главным образом потому, что разработчики не принимают во внимание навигационное измерение приложений. Естественно, они просто разрабатывают прикладную модель с объектами, а затем формируют интерфейс пользователя. Тем не менее существует много задач проектирования, которые можно решить, если только рассматривать навигацию как отдельную задачу проектирования.

Методология Relationship Management Methodology (RMM) первоначально была предложена Isakowitz в [4]. Она подходит для гипермедиа-приложений, имеющих неизменную структуру и динамические данные, требующие частого обновления, например каталоги продуктов. RMM плохо подходит для гипермедиа-приложений, которые являются хорошо структурированными, но с неизменными в течение продолжительных периодов данными, или гипермедиа-приложений с динамической структурой и динамическими данными. Основу этой проектной методологии можно условно разделить на три стадии.

Первая стадия — это представление информационной области гипермедиа-приложения посредством ER-диаграммы. Выделенные сущности и их отношения формируют основу гипермедиа-приложения, реализуемую как узлы и связи.

Вторая стадия — это определение информации для представления в узлах и уровнях доступа к ней пользователей, что подразумевает разделение сущностей на части для последующей группировки. Связи между частями называются структурными. Структурные связи отличаются от ассоциативных отношений в структурных связях, существующих между одной и той же сущностью, тогда как ассоциативное отношение ссылается на другие сущности. Это различие важно для навигации.

Третья стадия — это разработка навигации. Чтобы иметь возможность обновлять информацию, эта стадия описывается в терминах общих условий, а не жестко определяет сценарий навигации, что достигается ссылкой на свойства сущностей и их связей.

Общим во всех этих методах является то, что проектирование гипермедиа-приложений имеет такую же сложность, как и проектирование программного приложения общего назначения. ООНДМ [8], например, разделяет проектное пространство на четыре части: концептуальное проектирование, разработку навигации, абстрактную разработку интерфейса и реализацию. В каждой части должна быть построена проектная модель, что увеличивает время проектирования. Вообще ООНДМ обеспечивает известные объектно-ориентированные конструкции (например, классы, объекты, свя-

зи, сценарии использования, и т.п.) для определения моделей, что позволяет во время концептуального проектирования строить (объектно-ориентированную) модель проблемной области на основе этих конструкций и, таким образом, в принципе допускает модификацию разработанной концептуальной модели для другого потребителя, чем сокращает время проектирования. Однако это требует от разработчика дополнительных усилий.

Описанный в литературе опыт использования ООНДМ для проектирования дает возможность сделать некоторые выводы: метод позволяет лучше понимать проектные решения и получать подтверждение от пользователя, облегчая использование и развитие приложения, и одновременно улучшает связь между участниками проекта (менеджерами и конечными пользователями, проектировщиками и заказчиком). Тем не менее, если мы хотим улучшить разработку в смысле многократного использования, нам нужны абстракции более высокого уровня, чтобы подтверждать сложные проектные решения и записывать успешные решения для последующего использования.

### 3. ПРИМЕНЕНИЕ ШАБЛОНОВ В ПРОЕКТИРОВАНИИ WIS

Проектные шаблоны являются мощным средством для фиксирования и передачи проектного опыта. Они введены Alexander С. [1] в области градостроительства и широко используются в объектно-ориентированном проектировании программ [2]. Проектный шаблон описывает проблему и ее абстрактное решение таким образом, чтобы это решение можно было использовать для других экземпляров этой же самой проблемы, а также описывает разумное обоснование и выбор оптимального решения.

Гипермедиа — богатая область для обнаружения и применения шаблонов. Гипермедиа-шаблоны можно условно разделить на архитектурные, навигационные и шаблоны интерфейса. Архитектурные шаблоны решают общие проблемы (связь навигации с базами данных или программным обеспечением). Навигация и шаблоны интерфейса [6] так же, как образцы в [1], показывают возможности формирования информационного пространства. Они хранят коллективный опыт гипертекст-общества, использующего шаблоны как по одному [2], так и соединяя их в группы и в виде каталогов.

Навигационные шаблоны показывают, как расширить простую интерпретацию узлов и связей и перейти от обычных интуитивных решений к проблеме проектирования. Например, пока неопытные разработчики рассматривают ссылки как прочную семантическую связь, “Set-based

Navigation"-шаблон показывает, когда связь полезна для соединения узлов (например, когда они принадлежат одному множеству благодаря некоторым свойствам стоящей перед пользователем задачи). "Node in Context"-шаблон показывает, как разрешить ситуацию, когда один и тот же узел появляется в многочисленных множествах, допуская различные представления в зависимости от множества, в пределах которого он доступен.

Шаблоны интерфейса имеют дело с построением более "причудливых" интерфейсов. Например, "Information on Demand" описывает, как организовать объекты интерфейса, когда проектировщик хочет избежать слишком большой познавательной нагрузки, если узел имеет слишком много информационных пунктов для восприятия. Обнаружение новых шаблонов является полезной деятельностью, так как помогает увеличить свой опыт, повысить уровень рассуждений и помочь в оценке своего (и чужого) проекта. Языки шаблонов для пользовательских интерфейсов и проектного взаимодействия (множество шаблонов, которое покрывают целиком проектное пространство) только сейчас разрабатываются [9].

Шаблоны не привязаны к конкретному методу проектирования: можно использовать свой каталог шаблонов при разработке гипермедиа-приложения. Кроме того, шаблоны являются очень мощным средством, даже если не используется никакой метод проектирования, чтобы специфицировать гипермедиа-приложение.

Проектные шаблоны дополняют процесс проектирования, так как показывают решения, превосходящие полученные примитивными методами. Например, наивный объектно-ориентированный разработчик стремится строго следовать основным понятиям объектной парадигмы, изолируя структуру и алгоритмы в одном объекте. Тем не менее сложные проблемы требуют более гибкого решения, подобно шаблону Bridge, Strategy or State [2]. В этих случаях представление, алгоритмы или состояние объекта переносятся в объектное поле, определенное за пределами объекта в контексте отдельной иерархии. Мы можем определить шаблоны на уровнях другой абстракции; они действуют как микроархитектура в системе, улучшая системную модульность и снижая дополнения.

Когда мы создаем шаблон, то описываем, какие типы проблем порождают микроархитектуру, в которой шаблон должен быть использован, и что такое выбор оптимального решения при его использовании. Как объяснено в [2], шаблоны не являются ни исходными, ни новыми решениями программных проблем, но они хорошо показали себя в успешных проектах. Фактически, используя стандартные проектные методы (подобно OOHDM

или RMM) для работы со сложным проектом, хорошие разработчики применяют свой опыт, чтобы выделять повторяющиеся проблемы и использовать уже найденные решения. Это и есть момент появления шаблонов. Как таковые шаблоны не изобретаются отдельно от проблемы, а обнаруживаются в процессе проектирования. Часто опытным разработчикам эти шаблоны могут казаться очевидными, поскольку отражают обычно применяемые проектные решения. Но для менее опытных разработчиков эти шаблоны составят ценный источник экспертных знаний в области проектирования для создания нового проекта.

Проектные шаблоны имеют неопределимое значение с тех пор, как проектные знания не являются решением, ориентированным на настоящее, а являются проблемно-ориентированным решением. Отметим уникальное свойство шаблонов: они описывают проблемы, которые часто встречаются в процессе разработки, вместе с усилиями, связанными с решением этих проблем. Шаблоны имеют превосходные обучающие принципы, поскольку содержат контекст и описание выбора оптимального решения.

Проектные шаблоны полезны для записи повторяющихся проектных задач в программных системах. Изучение успешных приложений и абстракция проблем и решений помогают обнаруживать новые шаблоны, что увеличивает наше понимание процесса проектирования и, следовательно, мы можем находить общее у аналогичных приложений, устанавливая контекст, в котором эти шаблоны появляются.

Проектные методы должны развиваться в сторону включения новых шаблонов в наборы своих примитивов или создания абстракций.

## ЗАКЛЮЧЕНИЕ

Здесь была кратко описана проблема проектирования гипермедиа с точки зрения создания программного обеспечения. Гипермедиа-проектирование является важным вопросом в новых областях, таких как электронная торговля, цифровые библиотеки и т.п. Проектные методы сами по себе не являются панацеей. Взаимодействие гипермедиа и каталогов проектных шаблонов растет. Эти каталоги могут использоваться проектными методами для улучшения процесса разработки, обеспечивая удобный путь многократного использования проектов и/или отдельных компонент. Тем не менее взаимодействие между шаблонами и методами далеко от тривиального. Требуют исследования как процесс внедрения шаблонов в качестве примитивов новых методов, так и процесс определения систематиче-

ской стратегии понимания шаблонов и формализации однозначных нотаций для их записи.

### СПИСОК ЛИТЕРАТУРЫ

1. **Alexander C et all.** A Pattern Language. — N.-Y.: Oxford University Press, 1977.
2. **Gamma E. et all.** Design Patterns: Elements of Reusable Object-Oriented Software. — Addison-Wesley, 1995.
3. **Garzotto F. et all.** HDM — a model-based approach to hypertext application design // ACM Transactions on Information Systems (TOIS). — 1993. — Vol. 11(1). — P. 1—26.
4. **Isakowitz T. et all.** RMM: a methodology for structuring hypermedia design // Communications of the ACM (CACM). — 1995. — Vol.38, N 8. — P. 34—44.
5. ACM Hypertext '99 / Proc. Second Intern. Workshop on Hypermedia Development. Hypermedia Patterns / Ed. by D. Lowe et all. — Darmstadt, 1999.
6. **Rossi G. et all.** Design reuse in hypermedia applications development // Proc. of ACM Hypertext97. — Southampton, UK. — 1997. — P. 57—66.
7. **Schwabe D., Rossi G., Barbosa S.** Systematic hypermedia design with OOHDHDM // Proc. of the ACM Intern.l Conf. on Hypertext (Hypertext'96) . — Washington, 1996.
8. **Schwabe D., Rossi D.** An object oriented approach to Web-based application design //TAPOS (Theory and Practice of Object Systems) . — Wiley and Sons, 1998.
9. **Tidwell J.** Interaction design patterns //Patterns Languages of Programs (PLoP '98), Allerton Conference Center. — Urbana Champaign, 1998