

4. *Воронков Н.В.* Использование эвристических оценок в задаче автоматического реферирования текстов // Информационные системы и технологии: Тез. докл. 1-ой междунар. конф. Мн., 2002.

## **ВОПРОСЫ РЕАЛИЗАЦИИ КОМПОНЕНТ БИЗНЕС-ЛОГИКИ**

**А. Л. Дашкевич**

В данной статье рассматриваются вопросы применения наиболее перспективных информационных технологий и средств позволяющих автоматизировать процесс разработки WEB-приложений использующих платформу J2EE. Анализируются возможности применения аспектно-ориентированного подхода, для новых возможностей повторного использования кода. Приводится пример применения аспектного подхода для повторного использования компонента, реализующего шаблон Audit Log и его сравнение с другими реализациями.

Большинству разработчиков хорошо известна концепция «слоев» разделения сложного приложения на слабо связанные между собой модули [1, с. 47]. Такое разбиение позволяет в будущем более легкий переход при замене технологии в одном из «слоев» системы, делает систему гибкой к использованию новых технологий и средств. При разработке WEB приложений хорошо известна схема разбиения на «слои» представленная на рис. 1. На этом же рисунке для каждого из «слоев» описаны технологии, предоставляющие готовые каркасы, которые разработчик лишь должен наполнить необходимой логикой.

Нibernate представляет собой перспективный некоммерческий проект, который предоставляет каркас для отображения объектов в реляционную базу данных и язык манипулирования объектами транслирующийся в SQL [2].

Центральным компонентом типичного web приложения является бизнес уровень. Существует многообещающая технология Spring с открытым исходным кодом, которая позволяет определить, каким образом необходимо связывать бизнес объекты с остальными уровнями приложения [3]. Основной подход, который используется в Spring, это «Инверсия по управлению» (Inversion of Control), что делает его одним из наиболее гибких по управлению.

Spring имеет в своем ядре мощную поддержку АОП. Это позволяет по-новому посмотреть на повторное использование кода. Например, существует компонент (acegi-security), реализующий сквозную функциональность для аутентификации и авторизации пользователей в системе, базирующейся на Spring.

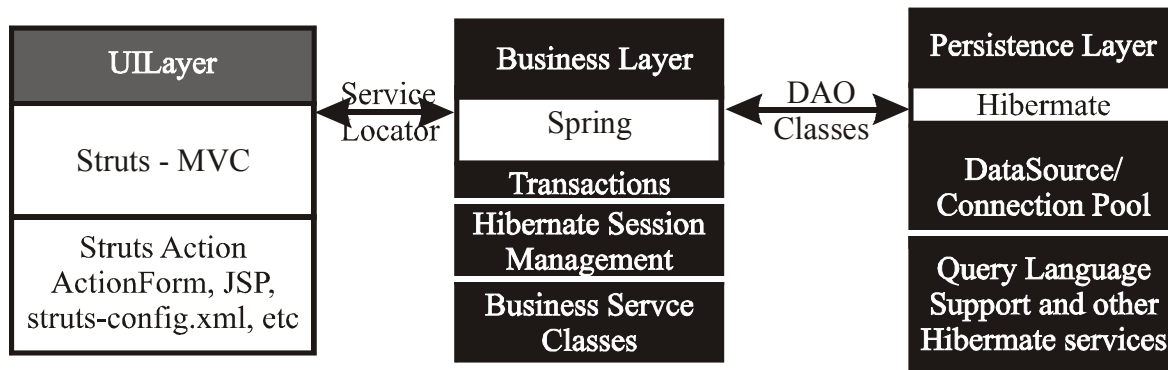


Рис. 1. Обзор технологий для каждого уровня приложения

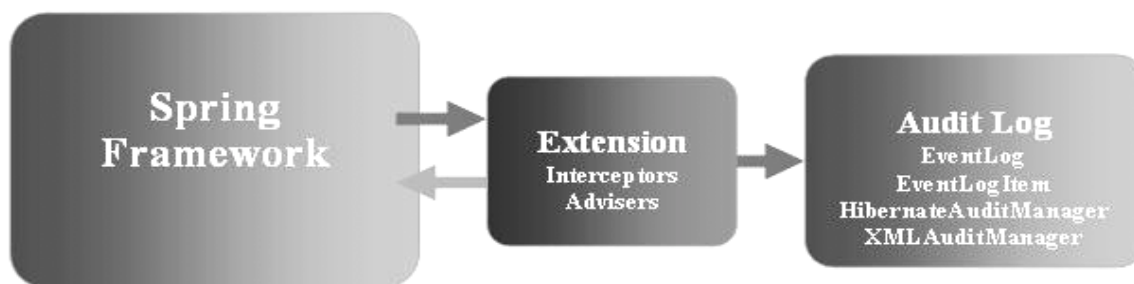


Рис. 2. Расширение ядра Spring для использования компоненты AuditLog

Ведение журнала событий также типичный пример сквозной функциональности, аспекта функционирования системы, реализация которого присутствует во многих программных модулях [4]. Поскольку реализация данной функциональности для Spring остается открытой, то была поставлена задача ее реализации (рис. 2).

В разработанном компоненте рассматривается аудит только следующих событий для бизнес объектов: сохранить (insert), обновить (update) и удалить (delete). Шаблон Audit Log реализован с помощью двух сущностей (EventLog и EventLogItem) связанных между собой отношением главный-подчиненный. Основная функциональность по управлению аудитом возложена на AuditManager. Этот базовый класс содержит основной метод audit, в который передается тип операции, объект операции, состояние и категория аудита. Наиболее сложной операцией является «обновление», поскольку нужно вычлениить все поля объекта, которые изменились. Краткая схема функционирования разработанного компонента представлена на рис. 3.

Разработанный компонент позволяет записывать информацию в базу данных посредством JDBC или Hibernate. Имеется возможность записи в XML формате, а также предусмотрено расширение для использования других способов хранения информации.

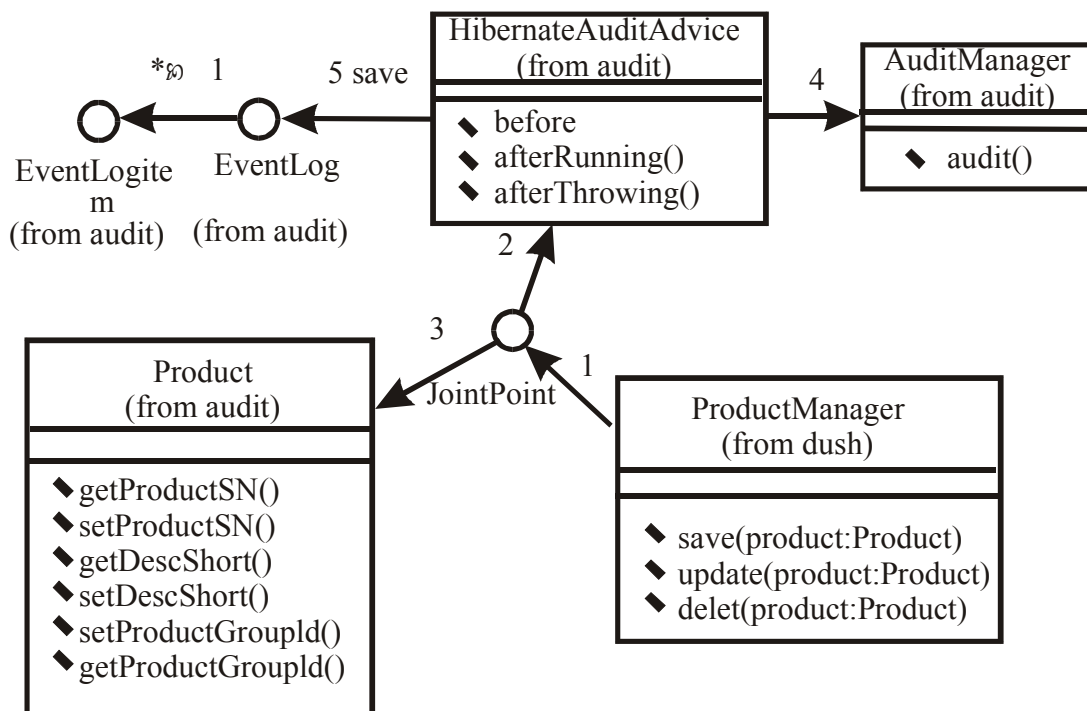


Рис. 3. Последовательность событий при аудите

После того, как необходимая информация записана, ее необходимо проанализировать. Обычно это очень сложно отследить изменения, если записей более тысячи. Аудит в XML файл предоставляет возможность применить XSL трансформацию для отображения информации в более читабельном формате.

В сравнении с другими подходами реализации шаблона Audit Log, использование АОП имеет ряд преимуществ:

- вынесение всей логики в один модуль,
- легкость конфигурирования,
- независимость от СУБД.

Audit Log редко реализуют в объектах бизнес логики, так как это приводит к довольно ощутимым затратам времени. Обычно этот шаблон реализуют с помощью хранимых процедур для СУБД. Этот подход, пожалуй, самый быстрый по времени выполнения аудита, однако и самый трудоемкий по разработке и конфигурированию, особенно если нужно поддерживать несколько СУБД.

### Литература

1. Фаулер, Мартин. Архитектура корпоративных программных приложений. М.: Издательский дом «Вильямс», 2004. – 544с.
2. <http://www.hibernate.org/>
3. <http://www.springframework.org/>
4. <http://www.javable.com/columns/aop/workshop/01/>