

Использование концепции надстраиваемых приложений для построения архитектуры модулей расширения портала корпоративной системы

К. Ю. Войтиков

Филиал Кемеровского государственного университета

в г. Анжеро-Судженске, Россия

e-mail: kost@asf.ru

А. Н. МОИСЕЕВ

Томский государственный университет, Россия

e-mail: amoiseev@kemsu.ru

This article is devoted to architecture of the components aimed at providing the extension service for an information portal of a corporate system. The proposed architecture allows extended functionality of a Web-view building subsystem without reconstruction of another parts of the system.

1. Постановка задачи

С развитием новых информационных технологий наблюдается повышенный интерес к компьютерным, а точнее, интегрированным системам, предназначенным для обеспечения эффективного управления предприятием в целом, — корпоративным информационным системам (КИС). Возникновение и внедрение в широкую практику служб сети Интернет определенным образом повлияло на технологию создания корпоративных информационных систем. Современную КИС трудно представить без возможности публикации данных ее информационных баз в веб. В связи с этим актуальными являются задачи, связанные с построением архитектуры системы формирования веб-представления в корпоративной информационной системе — информационного портала. Стоит сказать, что в работе под корпоративной информационной системой будут подразумеваться компьютерные системы масштаба предприятия среднего или малого размера. В отличие от больших корпоративных систем (построенных, например, на базе SAP R/3, Lotus Notes/Domino, Oracle и т. д.), в которых средства поддержки информационных порталов интегрированы в состав системы изначально, корпоративные системы среднего и малого класса не имеют подобных полнофункциональных средств.

Информационным порталом [1] будем называть рабочую среду, которая для пользователя увязывает воедино все корпоративные приложения и обеспечивает доступ ко всей информации и ресурсам посредством Интернет-технологий.

Для получения веб-представления данных в корпоративных системах иногда требуется предварительно преобразовывать информацию, полученную из источников данных предметной области. Такие задачи бывают не связаны ни с логикой предметной

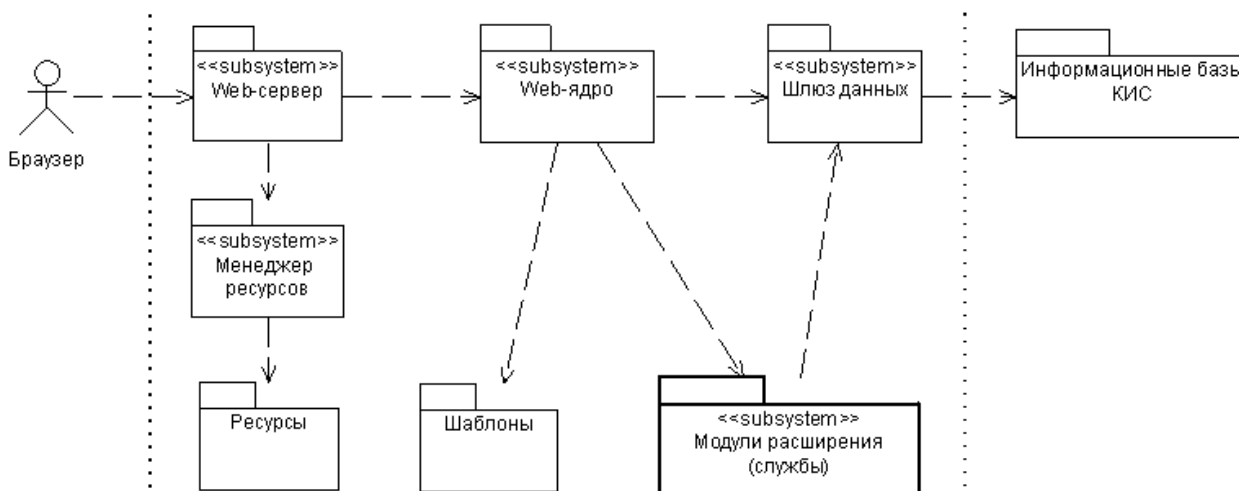


Рис. 1. Основные элементы архитектуры подсистемы формирования веб-представления

области, ни с механизмами формирования представления. В качестве примера можно привести модуль “Служба новостей”. Задачей такой службы является, например, компоновка объектов “Новость” (которые могут даже не существовать в исходной информационной базе) и формирование их списка, ограниченного по какому-либо признаку (по количеству или диапазону дат). Нетрудно заметить, что в данном случае ни одна из вышеизложенных задач не входит ни в логику слоя предметной области, ни в функции подсистемы формирования веб-представления. Для решения таких задач в работе предлагается использовать специальные компоненты — модули расширения (будем также называть их службами веб-портала). В задачи этих служб входит выполнение функций так называемой “логики представления” [2].

Поскольку подсистема формирования веб-представления должна поддерживать архитектуру клиент/сервер, а разработчики желают при этом владеть инструментом расширения функционала без перепроектирования системы в целом, актуальной является задача поиска подходящего решения. Предложения по построению архитектуры таких систем сформулированы в [3]. Представленная в этой работе концепция построения приложений позволяет проектировать программные системы клиент/сервер с архитектурой, устойчивой к покомпонентному расширению функционала.

Вопросы, связанные с построением архитектуры системы формирования веб-представления для корпоративных информационных систем, описаны в [1]. Согласно предложенному подходу, для решения данной задачи предлагалось возложить выполнение отдельных подфункций на соответствующие подсистемы. Основные элементы архитектуры представлены на рис. 1.

Предметом рассмотрения в данной работе является построение архитектуры модулей расширения информационного портала на основе концепции, изложенной в [3].

2. Реализация

Решение проблемы формирования “полноценного” представления на основе данных, полученных как из информационных баз предметных областей, так и от “модулей расширения”, заключается в применении архитектурного решения “Двухэтапное представление” (Two Step View) [2]. Первый этап формирования представления выполняют моду-

ли расширения, где строится “логическое представление”, не содержащее какого-либо специального форматирования, а второй — непосредственно само веб-ядро на основе шаблонов представления точно так же, как для объектов, полученных от “Шлюза данных”.

Согласно [1], веб-ядро формирует окончательное представление либо только на основе данных, полученных из предметной области, либо в комбинации с данными, синтезированными модулем расширения. Нетрудно заметить, что при построении веб-представления данные модуля расширений используются не всегда. Поэтому для работы с объектами, сформированными данным модулем, в подсистеме формирования веб-представления предлагается использовать паттерн “Заместитель” (Proxy) [4]. Использование такого решения позволит отложить расходы на создание и инициализацию необходимых объектов до момента, когда они действительно понадобятся. Для этого в подсистеме формирования веб-представления определим соответствующий класс — DocumentProxy. Данный класс хранит ссылку, которая позволит ему обратиться к реальной службе в случае необходимости, контролирует доступ к службам и отвечает за их создание и удаление. Представителем же реальной службы в подсистеме формирования веб-представления будут объекты класса PortalService. Предполагается, что в системе одновременно работающих служб несколько. Для обращения к ним создадим “библиотеку” — класс-контейнер служб (ServicesLibrary), отвечающий также за связь с подсистемой “службы”. Все эти классы показаны на диаграмме (рис. 2).

Для создания данной подсистемы воспользуемся концепцией построения архитектуры настраиваемых приложений клиент/сервер с обобщенным протоколом передачи данных [3]. При этом модули расширения будут являться серверами для подсистемы формирования веб-представления. Исходя из этого в модулях расширения определен класс LibraryController, являющийся потомком класса ServerController. Экземпляр-одиночка [4] этого класса отвечает за обработку запросов, поступающих от подсистемы формирования веб-представления.

В соответствии с архитектурным решением “Супертип слоя” (Layer Supertype) [2] также определим класс PortalService, являющийся абстрактным предком всех служб системы. В данном классе определим метод GetPackage(...), отвечающий за формирование пакета данных, созданных службой. Этот метод будет переопределен для каждой конкретной службы.

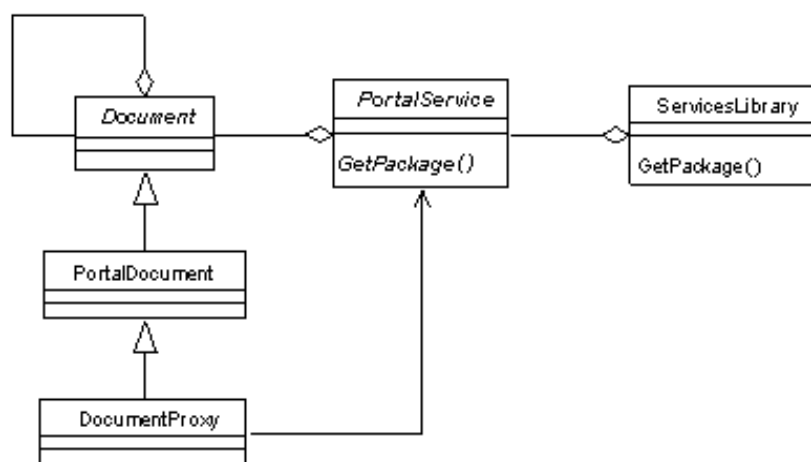


Рис. 2. Классы, обеспечивающие взаимодействие с модулями расширений

Опишем архитектуру модуля расширений на примере службы “Новости”. Задачами данной службы являются выборка по определенным признакам необходимых документов предметной области (новости и их содержание), преобразование их соответствующим образом (выделить заголовок, аннотацию, отсортировать и отфильтровать список), т.е. построение “логического представления”, на основе которого подсистема формирования веб-представления создает окончательное представление соответствующей страницы портала.

Необходимо обратить внимание на то, что результатом работы службы являются “виртуальные документы”, т.е. документы, не хранящиеся в базе данных, а сформированные динамически в зависимости от алгоритмов обработки. Действительно, страница новостей регулярно изменяется в зависимости от интенсивности ее пополнения и диапазона рассматриваемых дат, поэтому сформированный документ должен удовлетворять данным условиям. Каждая новость в отдельности представляет собой не что иное, как документ системы, а задача службы — выбрать необходимые (например, последние пять штук), преобразовать их и отправить сформированный список единым составным документом для его представления на веб-странице.

Для реализации модуля расширения “Новости” введем специальный класс `NewsMsg`, каждый объект которого представляет собой некоторую совокупность данных, достаточную для формирования одного сообщения-новости. Класс `NewsMsg` будет неизвестен подсистеме формирования веб-представления и доступен только внутри библиотеки модуля расширения.

Для реализации самой службы определим класс `NewsService`, унаследованный от класса `PortalService`. Объект данного класса будет отвечать за формирование пакета новостей на основе совокупности объектов `NewsMsg`.

Документы, которые передаются подсистеме формирования веб-представления, являются виртуальными, но для них также необходимо определить соответствующие классы:

`News` — документ, представляющий собой список новостей;

`OneNewsMsg` — документ, представляющий собой одну новость в списке.

В соответствии с этим диаграмма классов виртуальных документов “Службы новостей” будет выглядеть следующим образом (рис. 3).

После получения контроллером библиотеки запроса к службе новостей производятся инстанцирование класса `NewsService` и вызов его метода `GetPackage(...)`. В ходе

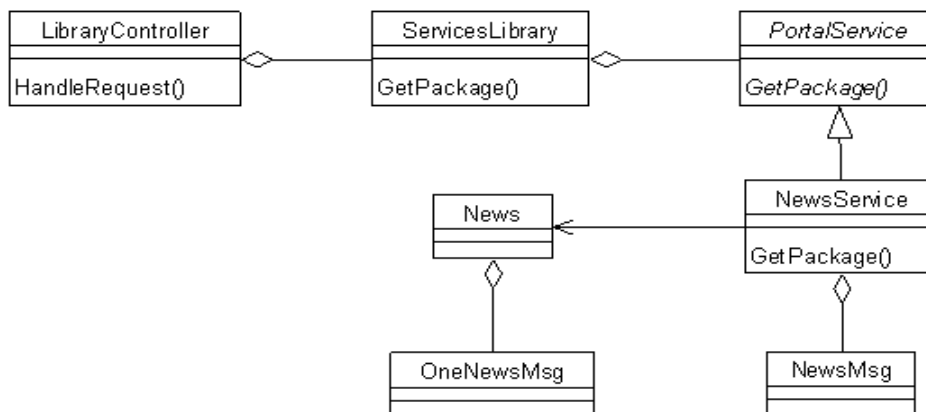


Рис. 3. Основные классы реализации службы “Новости”

работы данного метода создается список (объект класса News) подлежащих выводу на экран новостей. Далее выполняются выборка объектов NewsMsg и их проверка на соответствие критериям формирования списка. Для объектов NewsMsg, удовлетворяющих критериям, создаются соответствующие объекты класса OneNewsMsg, заполняются их атрибуты и объекты OneNewsMsg добавляются в список News. Затем (в случае необходимости) производится дополнительная сортировка этого списка.

Формирование пакета заканчивается вызовом метода GetPackage(...) для объекта-списка News.

При работе службы в составе корпоративной информационной системы в качестве “новостей” могут выступать различные документы прикладной области. Такими документами могут быть, например, кадровые приказы, поручения, отчеты и т. д. В этом случае работа “Службы новостей” будет производиться так же, как и прежде, только с учетом того, что в качестве объектов класса NewsMsg будут выступать объекты различных классов предметной области из соответствующих хранилищ.

Выводы

Применение модулей расширения для построения архитектуры системы формирования веб-представления обеспечит необходимый функционал системе в целом, сохраняя при этом способность к расширению и даже ее переориентации на новую предметную область, что означает высокую степень повторной используемости предлагаемого решения, а значит — его эффективность.

В предлагаемой архитектуре нет ограничения на уровне распределенности приложения, а применение концепции надстраиваемых приложений с обобщенным протоколом передачи данных [3] дает возможность подключать новые службы к подсистеме формирования веб-представления (веб-ядру) без ее перекомпиляции.

Таким образом, применение модулей расширения повышает гибкость использования сервисов, одновременно повышая степень их защищенности, модули расширения могут также выступать в роли сервисов веб-портала и использоваться внешними системами для удаленного доступа (например, формирование RSS-пакета).

Список литературы

- [1] Войтиков К.Ю., Моисеев А.Н. Типовая архитектура Web-портала корпоративной информационной системы // Вест. ТГУ. Приложение. 2006. № 19. С. 18–20.
- [2] ФАУЛЕР М. Архитектура корпоративных программных приложений: Пер. с англ. М.: Изд. дом “Вильямс”, 2004. 544 с.
- [3] Войтиков К.Ю., Змеев О.А., Моисеев А.Н., Якушев А.А. Архитектура надстраиваемых приложений клиент/сервер с обобщенным протоколом передачи данных // Вест. Том. гос. ун-та. 2004. № 284. С. 49–52.
- [4] ГАММА Э., ХЕЛМ Р., ДЖОНСОН Р., ВЛИССИДЕС ДЖ. Приемы объектно-ориентированного проектирования: паттерны проектирования: Пер. с англ. СПб.: Питер, 2001. 366 с.