

РАЗРАБОТКА ПРОГРАММНЫХ ИНТЕРФЕЙСОВ ДЛЯ РЕАЛИЗАЦИИ ДОКУМЕНТООБОРОТА В КОРПОРАТИВНОМ ХРАНИЛИЩЕ ЭЛЕКТРОННЫХ ДОКУМЕНТОВ ТПУ

Шерстнёв В.С., Иванов С.С., Макаров М.П.

ФГБОУ ВПО «Национальный исследовательский Томский политехнический университет», Томск, Россия (634050, г. Томск, пр. Ленина, 30), e-mail: vss@tpu.ru

В данной статье описывается процесс разработки программных библиотек для интеграции Java-приложений Oracle Portal и хранилища электронных документов на основе Oracle Universal Content Management. Сервер Oracle Universal Content Management позволяет осуществлять электронный документооборот и предоставляет для этого универсальный программный интерфейс, основанный на протоколе SOAP. Описываемое в статье разработанное программное обеспечение на языке Java предоставляет собой расширяемый высокоуровневый программный интерфейс для выполнения основных операций документооборота с использованием сервера UCM. В статье указано место разработанного UCM-адаптера как промежуточного звена между корпоративным порталом и хранилищем электронных документов, приведены этапы и особенности проектирования ряда модулей данного программного обеспечения, кратко описаны результаты выполненного тестирования и перспективы последующего внедрения.

Ключевые слова: электронные документы, корпоративный портал, корпоративное хранилище, Oracle UCM, Java, MVC, электронный документооборот, разработка приложения, ТПУ.

DEVELOPMENT OF THE PROGRAMMING INTERFACE FOR IMPLEMENTING OF ELECTRONIC DOCUMENTS FLOW BASED AT A CORPORATE DATA WAREHOUSE OF TPU

Sherstnyov V.S., Ivanov S.S., Makarov M.P.

National Research Tomsk Polytechnic University, Tomsk, Russia (634050, г. Томск, Lenin Avenue, 30), e-mail: vss@tpu.ru

In this article the process of developing software libraries for integration of java-software application based on Oracle Portal and electronic documents storage based on Oracle Universal Content Management is described. The Oracle Universal Content Management server enables electronic document flow and provides for this purpose the universal program interface based on the SOAP protocol. The developed software described in article in the java language provides itself the expanded high-level program interface for performance of the main operations of document flow with UCM server use. In article the place of the developed UCM adapter as intermediate link between a corporate portal and storage of electronic documents is specified, stages and features of design of a number of modules of this software are given, results of the executed testing and prospect of the subsequent of introduction are briefly described.

Keywords: electronic documents, corporate portal, corporate repository, Oracle UCM, Java, MVC, flow of electronic documents, application development, TPU.

Большинство современных предприятий постепенно переходит к использованию электронного документооборота. Преимущества электронного документооборота над традиционным «бумажным» достаточно очевидны и хорошо известны. Томский политехнический университет также формирует систему внутреннего электронного документооборота, интегрируя его с уже используемыми IT-решениями. В качестве основы построения системы электронного документооборота используется программный продукт Oracle Universal Content Management (Oracle UCM) [2]. В целом Oracle UCM является системой для управления электронным контентом предприятия, и на его основе разработано и запущено в пробную эксплуатацию хранилище электронных документов (ЭД)

университета [4]. Для контроля и упорядочивания поступления ЭД в хранилище требуется описать все существующие источники ЭД в университете и определить для них соответствующие «потоки документов». Под «потоком документов» здесь следует понимать перечень необходимых этапов и операций, предваряющих попадание документа в хранилище. Каждый поток ЭД описывается своим набором критериев (фильтром) приёма ЭД и маршрутом прохождения по ответственным лицам (экспертам). Основным источником внесения ЭД является корпоративный портал университета, построенный на основе Oracle Application Server. Следовательно, требуются соответствующие высокоуровневые программные интерфейсы, принимающие ЭД от программного обеспечения корпоративного портала и передающие их в электронное хранилище. Большинство программного обеспечения портала создано на языке Java, что рекомендует вести разработку необходимого дополнительного ПО тоже на этом языке. В процессе вложения ЭД в хранилище документ должен следовать в соответствующем «потоке документов». Схематично этот процесс представлен на рис. 1.

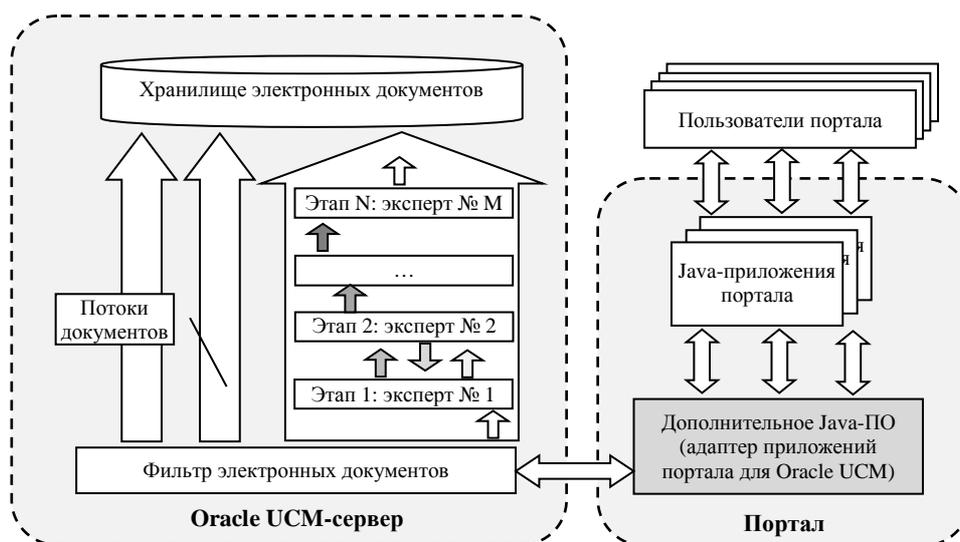


Рис. 1. Схема внесения электронных документов в корпоративное хранилище.

На рис. 1 показано, что все ЭД, вносимые пользователями корпоративного портала, в первую очередь диагностируются на соответствие тем или иным существующим потокам документов. После попадания в соответствующий поток ЭД циркулирует между этапами этого потока, пока не попадёт в корпоративное хранилище ЭД.

Описание решения задачи по разработке высокоуровневых программных интерфейсов, позволяющих упростить взаимодействие программного кода пользовательских интерфейсов (отвечающих за формирование пользовательского интерфейса) и программных интерфейсов хранилища ЭД, является основной целью работ, описываемых в данной статье.

Oracle UCM предоставляет определённый универсальный программный интерфейс для работы с ЭД и их потоками. Но, с точки зрения подходов к разработке сложных программных систем, прямое использование универсальных интерфейсов существенно увеличивает объём и сложность программного кода конечных пользовательских приложений на корпоративном портале. Более целесообразным видится создание слоя промежуточных высокоуровневых библиотек, предоставляющих порталным приложениям (приложениям, работающим под управлением Oracle Portal) более простой и удобный доступ к функциям по управлению электронным документооборотом. Промежуточный слой должен легко поддаваться масштабируемым изменениям с внесением минимальных корректировок в код; являться переносимым, то есть не зависеть ни от платформы использования, ни от веб-сервера; являться самостоятельным модулем.

Проектирование и разработка системы проходила согласно гибкой методологии разработки (Agile software development). Гибкими методологиями являются: XP, Scrum, Lean, Kanban, ScrumBut. На практике редко удается придерживаться какой-то конкретной методологии, чаще всего используется комбинация двух и более методологий. В рамках данной работы были частично использованы две гибких методологии разработки: XP и Scrum.

При этом общая архитектура разрабатываемых библиотек и тестовых пользовательских интерфейсов проектировалась с учетом архитектурного шаблона Model - View - Controller (MVC). Сутью данного шаблона является отделение бизнес-логики приложения от средств ее визуализации. Реализацией бизнес-логики в данном случае занимается система Oracle UCM и разрабатываемые высокоуровневые библиотеки. В качестве вариантов реализации средств визуализации информации может выступать веб-интерфейс или приложение для персонального компьютера, мобильного устройства или любой другой системы. «Моделью» (Model) в данном случае является набор документов в Oracle OCM, «Представлением» (View) являются пользовательские веб-интерфейсы, а «Контроллер» (Controller) должен представлять собой библиотеку с набором функций для манипулирования UCM-сервером, как показано на рис. 2. Ввиду того что функциональность может расширяться, библиотека контроллера должна быть открыта для расширений, но при этом закрыта для изменений уже существующего кода.

Согласно схеме, представленной на рис. 2, пользователь взаимодействует с неким пользовательским интерфейсом, который в свою очередь вызывает высокоуровневые функции разработанного адаптера к Oracle UCM. На основании принятой от пользовательского интерфейса информации адаптер формирует и выполняет соответствующие SOAP-запросы [8] к веб-сервисам Oracle UCM-сервера, расположенным на

этом же веб-сервере. Полученные от веб-сервисов XML-ответы декодируются соответствующим парсером и отображаются в пользовательский интерфейс (веб-интерфейс, интерфейс мобильного приложения или приложения персонального компьютера).

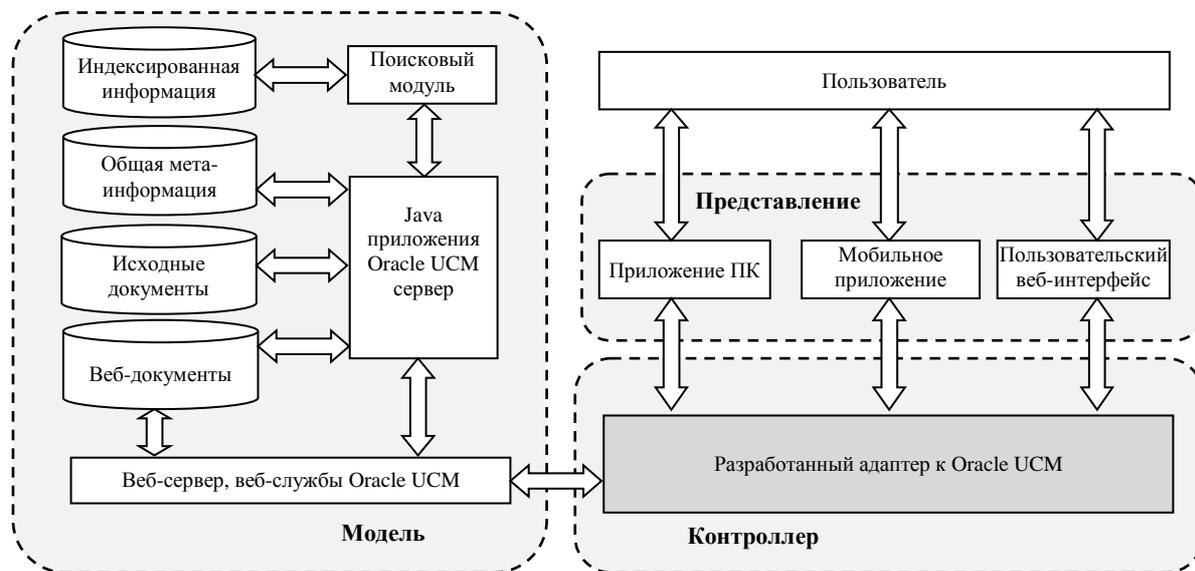


Рис. 2. Применение шаблона проектирования MVC к разрабатываемому программному обеспечению.

Адаптер к Oracle UCM условно можно разделить на несколько составляющих:

- интерфейсные классы, которые взаимодействуют непосредственно с UCM-сервером;
- анализатор XML-документов на основе реализации SAX-анализатора;
- фабрика классов для создания запросов на UCM-сервер;
- управляющий класс, который объединяет весь функционал адаптера.

Все запросы к UCM-системе представляют собой XML-документы. XML-документы используются при взаимодействии со всеми веб-службами сервера Oracle UCM. Для гибкой поддержки существующих XML-запросов к UCM-серверу и реализации возможности добавлять поддержку новых отправляемых запросов на UCM-сервер было решено использовать дополнительный шаблон проектирования «фабрика». Данный шаблон представляет собой независимые от приложения объекты вместе с зависимым от приложения объектом, которому они делегируют создание других, зависимых от приложения объектов. При этом требуется присутствие независимых от приложения объектов, которые иницируют операцию создания зависимых от приложения объектов, с предположением, что эти объекты реализуют общий интерфейс [3]. Таким образом, внешние системы будут работать с общим интерфейсом, но при этом создавать конкретные объекты классов. При этом класс-создатель не обладает информацией о фактическом классе создаваемого объекта.

Для выполнения задач, связанных с документооборотом, основанным на Oracle UCM, из разрабатываемой библиотеки адаптера потребовалось взаимодействовать с большинством

существующих веб-служб UCM, таких как: «GetWorkflowQueue», «GetWorkflowInfo», «WorkflowApprove», «WorkflowReject», «DocMetaData», «UserMetaData» и т.д.

Весь обмен информацией между веб-сервисами Oracle UCM и адаптером происходит по протоколу SOAP с обязательным использованием XML-документов, их составлением и анализом. Из трёх наиболее используемых подходов к обработке XML-документов (DOM, SAX, StAX) был выбран анализатор SAX (Simple API for XML) [7]. Данный анализатор обрабатывает XML-документ и вызывает методы, связанные с его различными участками, а основная программа, использующая анализатор, решает, как реагировать на тот или иной элемент XML-документа. SAX-анализаторы используются тогда, когда нужно извлечь информацию о нескольких элементах из XML-документа, либо когда информация из документа нужна только один раз [1].

В качестве основного средства разработки высокоуровневой библиотеки использовался язык Java и среда разработки JetBrains IntelliJ IDEA 10.0 [6].

В итоге, с учётом вышеизложенных проектных особенностей, были реализованы Java-библиотеки адаптера к UCM-серверу, позволяющие получить высокоуровневый программный интерфейс для работы с веб-сервисами сервера UCM. Реализованный функционал был собран в отдельные Java-модули (JAR-файлы, библиотеки), которые можно использовать в любом ином Java-проекте, где требуется взаимодействовать с Oracle UCM-сервером, как с сервером хранения ЭД.

Для проверки работоспособности созданных Java-библиотек адаптера к Oracle UCM был развернут тестовый веб-сервер с поддержкой Java-сервлетов (веб-сервер «Tomcat» [5]). На этот веб-сервер были перенесены JAR-файлы разработанного адаптера и созданы соответствующие тестовые пользовательские веб-интерфейсы, настроенные на использование адаптера.

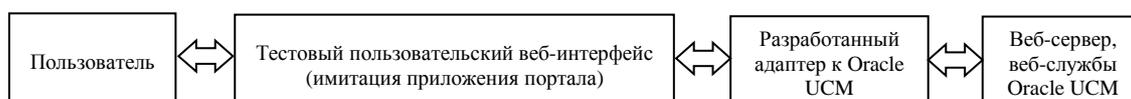


Рис. 3. Архитектура тестирования разработанного программного обеспечения.

Далее было проведено тестирование всех реализованных в адаптере функций взаимодействия с Oracle UCM. Разработанные функции успешно вносили документы на UCM-сервер, отображали документы для тех или иных экспертов этапов «поток документов», позволяли корректно перемещать документы по этапам «поток документов» и т.д. В итоге проведённого тестирования были выявлены и устранены незначительные неточности в работе разработанного программного обеспечения. После окончания стадии тестирования разработанных Java-библиотек предполагается их внедрение на корпоративный портал ТПУ.

Список литературы

1. Бекет Г. Java SOAP для профессионалов. – М. : Лори, 2012. – 456 с.
2. Управление контентом // Oracle Inc. : официальный сайт [Электронный ресурс]. - URL: <http://www.oracle.com/ru/products/middleware/content-management/index.html> (дата обращения: 01.11.2013).
3. Фримен Э. Паттерны проектирования. - СПб. : Питер, 2011. – 646 с.
4. Шерстнёв В.С., Иванов С.С., Акулин И.А. Использование Oracle Universal Content Management в качестве корпоративного хранилища документов ТПУ // Вестник науки Сибири : электронный научный журнал / Томский политехнический университет (ТПУ). – 2011. – № 1 (1). – С. 302-307.
5. Apache Tomcat // Apache Software Foundation [Электронный ресурс]. - URL: <http://tomcat.apache.org> (дата обращения: 01.11.2013).
6. JetBrains // IntelliJ IDEA [Электронный ресурс]. - URL: <http://www.jetbrains.com/idea> (дата обращения: 01.11.2013).
7. Megginson David. SAX // Official website for SAX [Электронный ресурс]. - URL: <http://www.saxproject.org> (дата обращения: 01.11.2013).
8. SOAP Version 1.2 Part 1: Messaging Framework (Second Edition) // World Wide Web Consortium [Электронный ресурс]. - URL: <http://www.w3.org/TR/soap> (дата обращения: 01.11.2013).

Рецензенты:

Ким В.А., д.т.н., профессор кафедры вычислительной техники Института кибернетики ФГБОУ ВПО «НИ ТПУ», г. Томск.

Авдеева Д.К., д.т.н., профессор, директор ООО «Медприбор», г. Томск.