

Организация информационных порталов на основе канальной интеграции*

С.П. Ковалев[†], К.Н. Яковченко[‡]

Аннотация. В докладе представлен новый подход к организации информационных порталов, основанный на канальной интеграции компонентов. Представлены преимущества этого подхода по сравнению с традиционными решениями. Рассмотрено его применение к построению распределенных приложений на основе технологий Grid.

Введение

Информационное сопровождение деятельности предприятий традиционно осуществляется с помощью веб-сайтов – хранилищ информации, поддерживающих доступ в интерактивном режиме. Однако специфика крупных предприятий или целых отраслей порождает потребность в более высокоорганизованных информационных системах, способных предоставить доступ к модулям автоматизации рабочих процессов в режиме реального времени. Здесь нужно обеспечить единую интегрированную точку входа в распределенную совокупность гетерогенных ресурсов предприятия, с сохранением привычного для пользователей сайтов стиля веб-интерфейса. Программная система, обслуживающая такую точку входа, называется информационным порталом [1].

Следует подчеркнуть, что в задачи портала не входит дублирование функций автоматизированных систем управления предприятием. Поэтому качество сервиса (quality of service), предоставляемого порталом, определяется в основном нефункциональными характеристиками – эффективностью использования ресурсов, эргономичностью пользовательского интерфейса, удобством конфигурирования и сопровождения, легкостью интеграции с другими приложениями. Разработка портала в основном сводится к сборке и настройке готовых интерфейсных компонентов, взаимодействующих через общую интеграционную среду, которая предоставляется технологической платформой порталного решения. Концептуальные подходы к организации такой среды, а также к созданию распределенных приложений на ее основе, составляют предмет рассмотрения настоящего доклада. Представлена методика канальной интеграции компонентов, позволяющая создавать порталы, отличающиеся высоким уровнем качества сервиса.

*Работа выполнена при финансовой поддержке Программы Президиума РАН № 21 “Разработка фундаментальных основ создания научной распределенной базы информационно-вычислительной среды на основе технологий GRID”.

[†]Институт вычислительных технологий СО РАН.

[‡]Новосибирский филиал ФГУП “ЦНИИАТОМИНФОРМ”.

1. Интеграция компонентов информационного портала

В настоящее время интеграцию компонентов информационного портала традиционно осуществляют путем помещения их в специальную программную среду – контейнер, предоставляющий службы системного уровня [2]. Основными задачами контейнера являются обеспечение удаленных вызовов, управление жизненным циклом компонентов, управление транзакциями, обеспечение безопасности. Благодаря этому отдельным компонентам не приходится заботиться об управлении ресурсами – они делегируют соответствующие функции контейнеру. Наиболее широко известной технологией контейнерной интеграции является Enterprise Java Beans (EJB) [3]. Компонентная модель EJB удовлетворяет спецификации Java 2 Enterprise Edition (J2EE), которая является де-факто промышленным стандартом, регламентирующим организацию корпоративных приложений на базе технологии Java. К числу лидеров среди порталных решений, основанных на этих технологиях, относится WebSphere Portal от компании IBM [4]. Основной единицей интеграции в этом продукте является портлет – приложение J2EE, которое обрабатывает данные запросов, приходящих с web-страниц пользователей, и выдает результаты также в виде web-страниц. Портлеты могут выполнять разнообразные функции, от отображения статического web-документа до реализации целого корпоративного приложения. Активизация и обслуживание портлета, реализующего функцию, затребованную в запросе, выполняется контейнером, входящим в сервер портала WebSphere.

Ориентация на обслуживание каждого запроса по отдельности, лежащая в основе контейнерного подхода к организации портала, является источником ряда трудностей. Дело в том, что рабочие процессы предприятия, для информационного сопровождения которых предназначен портал, как правило, имеют структуру последовательностей типовых шагов, выполняемых пользователями, причем параметры очередного шага определяются контекстом исполнения процесса (а также текущим состоянием предприятия в целом). Однако ни контейнер, ни содержащиеся в нем функциональные компоненты не приспособлены для отслеживания этого контекста. Поэтому распределение ресурсов приходится осуществлять исходя из потенциально возможных “наихудших” случаев, а не из реальных потребностей. Ярким примером служат ресурсы, ориентированные на установление логического соединения – например, доступ к реляционным базам данных. Контейнеру приходится постоянно поддерживать большой набор (пул) соединений, направляя запросы от компонентов в свободные соединения по мере поступления. Емкость пула, как правило, существенно превышает фактическую плотность одновременно выполняемых запросов, что вызывает нерациональный расход ресурсов. Действия, направленные на освобождение ранее выделенных ресурсов, в том числе завершение работы экземпляров компонентов, выполняются в зависимости от показателей системного характера (например, отсутствие обращений в течение определенного промежутка времени), а не от состояния сопровождаемых рабочих процессов.

Для преодоления этих трудностей авторами разработан альтернативный принцип организации информационного портала, названный канальной интеграцией. Здесь интеграционная среда не ограничивается диспетчеризацией за-

просов, а создает и поддерживает логические каналы, связывающие элементы пользовательского интерфейса с функциональными компонентами. Возможно также построение межкомпонентных каналов, по которым передаются внутренние рабочие данные. С каналом ассоциируется информация о структуре и ходе выполняемого рабочего процесса, и системные ресурсы выделяются исходя из его потребностей, а также из настроек пользователя (причем пользователь может иметь различные права в различных каналах). Кроме того, каналы способны выполнять различные кросс-компонентные функции, такие как трансляция данных (например, перевод между американскими единицами измерения и метрической системой), которые в контейнерном подходе не удастся вынести за пределы компонентов. Особо следует отметить возможность интеллектуального протоколирования актов обмена данными, проходящих по каналам, с отражением распределения по рабочим процессам и идентификацией “смысла” передаваемых данных, определяемого контекстом.

Отличие канального подхода к интеграции портала от контейнерного можно проиллюстрировать следующим образом. Основной задачей портальной платформы является предоставление магистральной среды передачи рабочих данных “высокого уровня”, оформленных в виде web-документов, наборов записей из реляционной базы (record sets) и т. д. Такую среду иногда называют бизнес-шиной. Как и обычные сети передачи данных, она может коммутировать либо отдельные пакеты (что соответствует контейнерной интеграции), либо каналы между конечными пунктами обмена данными (как в канальном подходе). Среда канальной интеграции отличается более точным соответствием между выделяемым и фактически используемым объемом ресурсов, сравнительно высоким уровнем “интеллекта”, возможностью быстрой поставки базовой конфигурации с минимальным набором важнейших функций. В целом, можно говорить о повышении уровня качества сервиса портала путем применения принципов канальной интеграции.

С помощью канального подхода было построено несколько порталов, в том числе портал информационного сопровождения деятельности атомной отрасли Российской Федерации www.minatom.ru. Здесь основной единицей информации, предоставляемой пользователю, является материал, состоящий из выходных данных, анонса, тела и присоединенных файлов. Реализованы каналы для доставки материалов, голосования, рассылки буклетов и т. д., которые обеспечивают индивидуальное распространение каждого информационного блока от автора через редакцию к подписчикам канала, с учетом их настроек. В рамках портала поддерживается распределенная корреспондентская сеть с отдельной редакцией на каждый канал.

2. Технологический процесс организации портала

С технологической точки зрения разработка портала является классическим частным случаем задачи компонентного проектирования. В таких задачах объем трудозатрат, связанных непосредственно с программированием, сравнительно мал. Первостепенное значение приобретают анализ и разделение ключевых ответственностей (separation of concerns), возлагаемых на компоненты, а также унификация правил их взаимодействия с окружением. Компонен-

ты (components) являются элементарными архитектурными абстракциями, с которыми оперирует разработчик компонентных систем. Посредством связей (connectors) они соединяются в целостные конфигурации (configurations) с учетом ограничений (constraints), в том числе нефункциональных [5].

Технологическая среда построения порталов является носителем связей, так что выбор модели ее организации является ключевым проектным решением, которое следует принимать исходя из результатов анализа требований. Естественным критерием адекватности такого выбора служит возможность явной привязки (трассировки) исходных требований к архитектурным единицам. Например, если строится портал, предназначенный для выполнения большого количества слабо связанных функций, то требования следует оформлять в виде вариантов использования, и далее реализовывать с использованием контейнерного подхода. Если же процессы, происходящие в сопровождаемой предметной области, подчиняются ограниченному числу многошаговых типовых сценариев, то здесь больше подойдет сценарный анализ и последующее применение канальной модели. Именно такая ситуация чаще всего имеет место в задачах информационного сопровождения деятельности предприятий и отраслей.

Исходя из результатов анализа сценариев, проектируется нужная конфигурация компонентов и каналов. Как обычно, многие каналы оказываются асимметричными – одна из конечных точек находится в режиме ожидания и последующей обработки запросов, приходящих от второй. Такие каналы отражают традиционную модель организации сценария “клиент-сервер”. Однако эта модель не является обязательной в общем случае – обе стороны могут проявлять спонтанную активность, для поддержки которой канал предоставляет средства асинхронного согласования режима взаимодействия. Кроме того, для реализации сложных информационных отношений строятся многоточечные каналы, обладающие возможностью собирать (мультиплексировать) данные из нескольких источников и дублировать их в несколько назначений.

Реализацию каналов и компонентов можно выполнять на базе широко распространенных технологий разработки распределенных систем, как J2EE или .NET. Это позволяет использовать готовые реализации отдельных компонентов от сторонних производителей, выполненные на этих технологиях, а также упрощает включение в проект новых разработчиков. Среда канальной интеграции строится на основе серверов приложений (application servers), которые хорошо приспособлены к функционированию в режиме промежуточного слоя между web-клиентами и источниками корпоративных данных. Хотя традиционный для web-технологий режим взаимодействия с клиентами, состоящий в обмене отдельными запросами, здесь сохраняется из соображений совместимости, обработка запросов внутри порталной среды устроена согласно принципам канальной интеграции. В описанном выше портале www.minatom.ru использовался сервер приложений Resin.

Использование канального подхода позволяет существенно уменьшить время поставки создаваемого портала. Как показал опыт, при наличии определенного набора готовых базовых компонентов первая работоспособная конфигурация, демонстрирующая поддержку ключевых рабочих процессов (возможно, с “заглушками” вместо некоторых операций), может быть собрана в течение 1–2 дней. При этом не требуется предварительно проводить такие “тяжело-

весные” мероприятия, как реструктуризация рабочей среды предприятия или замена всех унаследованных систем, не пригодных к интеграции в рамках выбранного порталного решения. Дальнейшее наращивание функциональных и презентационных возможностей портала также можно осуществлять небольшими шагами, на каждом из которых добавляется новый канал, компонент или кросс-компонентная функция. Реализацию шагов, слабо зависящих друг от друга, легко организовать параллельно, задействуя небольшие независимые группы разработчиков.

Существенно, что результаты каждого шага хорошо видны пользователям – у них появляется поддержка нового процесса или операции. В этих условиях легко обеспечить постоянный контроль над ходом разработки со стороны потребителя – если результаты очередного шага не удовлетворяют пользователей или произошло изменение требований к нему, то его можно скорректировать или вообще отменить без особого ущерба для календарного и финансового плана проекта в целом. Технологический процесс приобретает положительные черты, свойственные адаптивным подходам к разработке программных систем, таким как экстремальное программирование [6]. В дальнейшем многие действия, связанные с усовершенствованием портала, удастся полностью передать сопровождающему персоналу. Таким образом, подход к организации информационного сопровождения деятельности, основанный на управлении каналами, отличается эффективностью и эргономичностью.

3. Интеграция компонентов и технологии Grid

Важной сферой применения методов интеграции компонентов является разработка распределенных приложений на основе технологий Grid [7]. Эти технологии предназначены для объединения гетерогенных вычислительных ресурсов, находящихся в различных административно-технических условиях, в общие глобальные сети. Традиционно к таким ресурсам относят процессорное время, память, хранилища данных. Однако сами по себе они не представляют большой ценности в силу зависимости от архитектуры управляющих ими аппаратных средств и операционных систем, а также из-за динамического характера показателей их доступности. С другой стороны, процесс выполнения сложного вычислительного приложения складывается из шагов, каждый из которых заключается в решении вполне конкретной задачи: применение разностной схемы, фильтрация массива данных и т. п. Поэтому на самом деле интерес представляют ресурсы высокого уровня, оформленные как программные компоненты, путем интеграции которых можно строить Grid-приложения [8].

Технологическую основу для компонентной сборки Grid-приложений представляет архитектура Open Grid Service Architecture (OGSA) [9]. В рамках этой архитектуры ресурс представляется как Grid-сервис – web-сервис, интерфейс программного доступа к которому удовлетворяет определенным соглашениям, обеспечивающим удобное автоматическое обнаружение и управление. Таким образом, достигается преимущество Grid-технологий по отношению к web-технологиям, что позволяет применять методики web-интеграции компонентов при разработке высокопроизводительных распределенных вычислительных приложений.

К числу таких методик относится описанная выше интеграция информационных каналов. Действительно, функционирование Grid-приложения можно представить как поток актов обмена данными между Grid-сервисами, представляющими вычислительные компоненты, по связывающим их каналам. При этом в Grid-сети ассортимент и характеристики компонентов и каналов меняются с течением времени, в зависимости от текущего профиля загрузки ресурсов. Конфигурация каналов формируется динамически, поэтому функции архитектора компонентного приложения нужно выполнять постоянно. Для этого требуется автоматический планировщик – кросс-компонентная служба, отвечающая за эффективность Grid-приложения. Благодаря простоте и технологичности процедуры изменения конфигурации путем перекоммутации каналов, такие планировщики, отличающиеся малым временем отклика на изменение профиля загрузки ресурсов, действительно могут быть созданы. Отметим, что поскольку планировщику приходится решать сложную оптимизационную задачу, ему можно задавать различные “подсказки”, например, в форме рекомендуемого шаблона развертывания (deployment pattern) [10].

Дополнительным преимуществом разработки Grid-приложений на основе канальной интеграции является возможность построения информационного портала приложения, который предоставляет пользователям автоматизированное рабочее место. С помощью этого портала осуществляется взаимодействие с компонентами в интерактивном режиме, мониторинг хода выполнения приложения, а также (при наличии административного уровня доступа) ручная подстройка конфигурации задействованных ресурсов. При использовании описанной выше технологии канальной интеграции такой портал формируется практически “бесплатно”, из готового набора элементов пользовательского интерфейса к динамической конфигурации информационных каналов.

Заключение

В докладе представлен новый подход к организации информационных порталов, основанный на канальной интеграции компонентов. По сравнению с традиционным методом интеграции с помощью контейнеров, этот подход позволяет более точно предсказывать потребности компонентов в ресурсах, упрощать реализацию кросс-компонентных функций, повышать уровень интеллектуальности при мониторинге потоков рабочих данных. При этом удается сохранить совместимость с существующими решениями. Разработка такого портала сводится к быстрой инкрементной поставке по мере готовности поддержки отдельных рабочих процессов. Адаптивный характер технологии канальной интеграции позволяет применять ее также при построении распределенных вычислительных приложений на основе технологий Grid.

Список литературы

- [1] Надеин А., Кузнецов В. Корпоративные Интранет-порталы. – М.: TopS Business Integrator, 2003. – <http://www.e-commerce.ru/analytics/analytics-part/analytics15.html>.
- [2] Цимбал А.А., Аншина М.Л. Технологии создания распределенных систем. – СПб.: Питер, 2003.

- [3] Monson-Haefel R. Enterprise Java Beans: 3rd ed. – Sebastopol: O'Reilly, 2001.
- [4] Credle R. IBM WebSphere Portal for Multiplatforms V5 Handbook. – IBM, 2004. – <http://www.redbooks.ibm.com/redbooks/pdfs/sg246098.pdf>.
- [5] Shaw M., Garlan D. Formulations and formalisms in software architecture // Computer Science Today. Lect. Notes in Comput. Sci. – 1996. – Vol. 1000. – P. 307–323.
- [6] Астелс Д., Миллер Г., Новак М. Практическое руководство по экстремальному программированию. – М.: Вильямс, 2002.
- [7] Foster I., Kesselman C. The Grid: Blueprint for a New Computing Infrastructure. – Morgan Kaufmann Publishers, 1999.
- [8] Furmento N., Mayer A., McGough S., Newhouse S., Darlington J. A component framework for HPC applications // Lect. Notes in Comput. Sci. – 2001. – Vol. 2150. – P. 540–548.
- [9] Foster I., Kesselman C., Nick J.M., Tuecke S. The Physiology of the Grid. An Open Grid Services Architecture for distributed systems integration // Grid Computing: Making the Global Infrastructure a Reality. – New York: Wiley & Sons, 2003. – P. 217–250.
- [10] Enterprise Solution Patterns Using Microsoft .NET. – Redmond: Microsoft Corp., 2003. – <http://msdn.microsoft.com/architecture/patterns/>.