

# Об организации доступа к данным в Информационном web-портале

© А.В. Босов, А.Н.Полухин

Институт проблем информатики Российской академии наук  
[AVBosov@ipiran.ru](mailto:AVBosov@ipiran.ru), [APoluhin@ipiran.ru](mailto:APoluhin@ipiran.ru)

## Аннотация

Представлено решение, включающее технологию и практическую реализацию, использованное в Информационном web-портале для поддержки контента, поставляемого в портал разнородными внешними информационными источниками. Предложенная архитектура сервера интеграции и доступа основана на концепции функциональной интеграции внешних систем и развитии технологии адаптеров.

## 1 Введение

Проект Информационного web-портала – собственного порталного решения, создаваемого в рамках программы информатизации РАН – успешно развивается уже в течении нескольких лет [1-3]. Одна из основных причин принятия решения о создании собственного порталного решения состояла в том, что имеющиеся коммерческие продукты не обладают достаточной гибкостью при работе с разнородной информацией. Учитывая то, что под порталом понимается система, обеспечивающая удобный сервис при работе с множеством самостоятельных информационных источников, образующих распределенную федеративную среду, возможности, предоставляемые порталом в этой части, выходят на первый план.

Концепция единой точки доступа к распределенным разнородным данным – основа любой порталной технологии. В частности, «вертикальным порталом» как раз и называют web-систему, обеспечивающую доступ и манипулирование корпоративной информацией через единую точку входа [4]. Другой вопрос, как разработчики реализуют эту концепцию в конкретных решениях. В большинстве известных решений, по крайней мере, на концептуальном уровне, используются так называемые «адаптеры», обеспечивающие преобразование дан-

ных неких внешних систем в форматы, «понятные» portalу. Такая, несомненно, упрощенная формулировка отражает суть, детализируемую уже конкретным решением. Так, портал Microsoft Sharepoint активно использует интеграцию с офисными приложениями на основе технологии web-папок, позволяющей максимально упростить коллективную работу с офисными документами. Для подключения других источников используются готовые или создаваемые с помощью предлагаемых API компоненты (web-parts). В портале Oracle основное внимание уделяется управлению web-контентом. Этот портал опирается преимущественно на хранение данных во внутреннем формате, в который должны импортироваться данные внешних информационных источников. В связи с этим внутренне хранилище организовано достаточно гибко, чтобы принимать данные с различной структурой, набор атрибутов для хранящихся объектов может необходимым образом корректироваться. В портале IBM WebSphere акцент сделан на интеграцию приложений. Данные получают непосредственно из интегрируемых приложений, для чего используются специальные портлеты (готовые или создающиеся с помощью предлагаемых API), трансформирующие внутренние запросы портала в обращения к соответствующим приложениям. Для поддержки подобного режима используется развитый программный интерфейс портлетов. Портал Hummingbird ориентирован на интеграцию данных в масштабе корпорации, в т.ч. из территориально распределенных источников. В основе механизма интеграции лежит использование централизованного репозитория метаданных. Компоненты ядра механизма интеграции обеспечивают управление процессами обмена и преобразования данных на основе сохраняемых в репозитории метаданных. Акцент делается на использовании возможностей обмена и преобразования данных, реализованных в конкретных источниках данных.

В общем случае можно считать, что подход порталов к интеграции внешних систем таков: один адаптер – одна внешняя система, внутри же портала информация представляется в некотором унифицированном внутреннем формате. Таким образом, если какому-то порталному приложению требуется работать с определенным источником, то он полно-

стью зависит от адаптера этого источника. В целом такой подход представляется вполне состоятельным, но порождает определенные сложности, попытка преодолеть которые или, по крайней мере, минимизировать их предложена в Информационном web-портале.

Предлагаемый подход основан на уточнении понятия «адаптер». Именно, в Информационном web-портале работа адаптера регламентируется требованиями поддержки строго определенного функционального интерфейса. Именно, предоставляемый адаптером программный интерфейс является фиксированным набором вызовов функций с параметрами, так что каждому возможному запросу к внешней системе соответствует в точности один вызов функции. Такой подход, с одной стороны, конечно, ограничивает возможности взаимодействия с информационным источником. С другой стороны, как показано далее, существенно облегчает реализацию порталного решения в целом и расширяет возможности интеграции поставляемого разнородными системами контента. Практически для любой системы, заинтересованной во взаимодействии с порталом, можно с минимальными затратами разработать адаптер портала, не внося принципиальных изменений в функционал самой системы, и подключить «на ходу» к portalу. При этом есть, конечно, набор готовых (стандартных) адаптеров.

Набор функций, которые должен реализовывать адаптер, формируется достаточно гибко, позволяя реализовать как самые простые схемы взаимодействия (например, полнотекстовой поиск по ресурсу), так и довольно сложные (вплоть до объединения схем репозитория источника и портала). Принципиальным для формирования этого набора является требование унифицировано работать с источниками как с хранилищами данных, т.е. посредством функционального интерфейса осуществлять доступ и манипулирование данными внешней системы.

## 2 Инфраструктура портала

В существующей реализации Информационного web-портала имеется достаточно развитая программная инфраструктура, представляющая собой ряд взаимодействующих служб. Рассматриваемая в данной работе служба, обеспечивающая взаимодействие внешних информационных источников, названа *сервером интеграции и доступа* к данным. Для работы этого сервера наиболее существенны следующие порталные подсистемы:

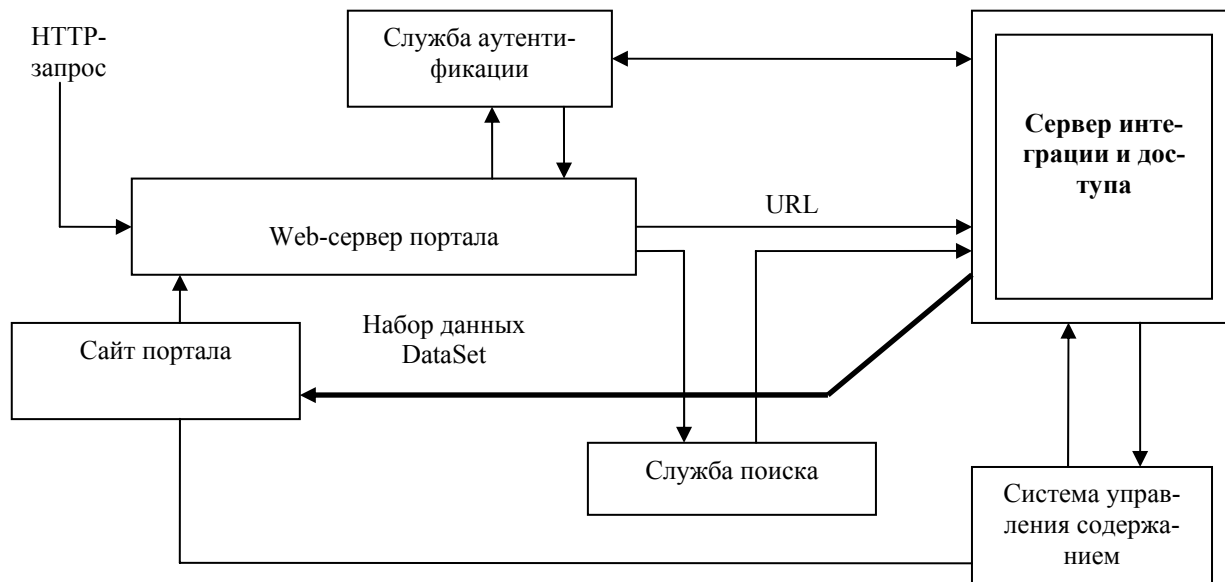
- сайт и система управления содержанием;
- подсистема безопасности (прежде всего сервис аутентификации);
- служба поиска;
- адаптеры.

*Сайт портала и система управления содержанием* в конечном итоге обеспечивают отображение данных пользователю (и форм манипулирования этими данными). Таким образом, соответствующим службам необходимо иметь возможность извлекать

данные из поставщика информации. Единственным поставщиком информации для этих служб является сервер интеграции и доступа. Сами службы «не знают», откуда и как данные извлекаются, просто любой код для манипуляции с данными обращается, используя единственный унифицированный механизм, к серверу интеграции и доступа, который выполняет запрос и возвращает результат. Причем возвращаемый результат также унифицирован: им всегда является реляционный набор, представляемый объектом DataSet используемой платформы .Net.

*Сервис аутентификации* обеспечивает идентификацию пользователей в портале и является основной частью подсистемы безопасности портала. Согласно идеологии обеспечения безопасности в Информационном web-портале этот сервис обеспечивает гарантированную идентификацию пользователя и привязку к сессии атрибутов, необходимых для реализации ролевой модели разграничения доступа, собственно же авторизацией должен заниматься сам информационный источник. Взаимодействуя с сервисом аутентификации, сервер интеграции и доступа может как сам обеспечить авторизацию (для чего используются декларативные ограничения, накладываемые на описания каждой функции для каждого адаптера, подключенного к серверу), так и передать необходимые атрибуты информационному источнику (внеся их в предусмотренные параметры в вызовах функций адаптера) для самостоятельного решения вопросов авторизации.

*Служба поиска* обеспечивает формирование полнотекстовых и атрибутивных запросов пользователей портала на поиск данных в информационных источниках для поиска, маршрутизацию поисковых запросов, сбор и консолидацию результатов поиска. Для тех внешних систем, что взаимодействуют с порталом посредством адаптеров, подключенных к серверу интеграции и доступа, последний является единственным источником, используемым службой поиска. Этим не ограничивается взаимодействие со службой поиска. Именно, службой поиска ведется специальная поисковая подсхема в рамках виртуальной (канонической) схемы данных портала. Собственные сервисы службы поиска позволяют формировать эту схему, подключая к ней различные адаптеры «в обход» сервера интеграции и доступа. Для таких, не подключенных к серверу интеграции и доступа, информационных источников для этого нужен пользователь, выполняющий по сути роль администратора внешней системы. В случае же, когда источник подключен к серверу интеграции и доступа, формирование поисковой подсхемы выполняется автоматически. При этом та часть поисковой подсхемы, которая соответствует серверу интеграции и доступа (а следовательно, и всей совокупности внешних систем, подключившихся к portalу через этот сервер), непосредственно извлекается из схемы данных сервера интеграции и доступа, выполняющей помимо прочего функцию виртуальной схемы репозитория портала (см. рисунок).



### 3 Место сервера интеграции и доступа в архитектуре портала

Конечной целью любого портала является взаимодействие с пользователем посредством web-интерфейса. Ради этого и решаются задачи, связанных с наполнением этого интерфейса данными. Любой код в портале, таким образом, должен быть разделен на две части: одна часть решает задачи получения/передачи собственно данных, другая – интерфейсные вопросы. Множественность информационных источников свидетельствует о сложности задачи получения/передачи данных, поэтому обоснованным представляется выделение этой задачи в качестве самостоятельной. Выделив задачу обмена данными между информационными источниками и порталом в самостоятельный блок, всему остальному коду портала обеспечивается независимость от информационных источников, т.е.

- независимость от способа взаимодействия с источником (от конкретного программного интерфейса адаптера внешней системы);
- независимость от форматов данных, принятых в источниках;
- независимость от способа хранения данных в источнике;
- независимость от расположения источника (т.е. одинаковые условия выполнения и в локальной, и распределенной средах, абстрагирование от таких параметров как адреса, URL, и т.п.).

Перечисленным требованиям и должен удовлетворить сервер интеграции и доступа.

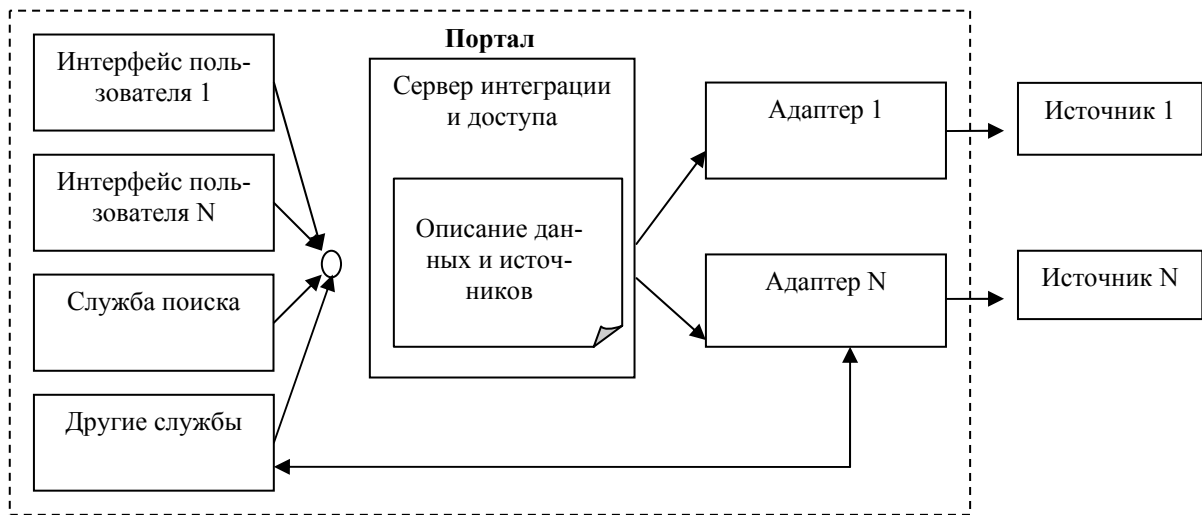
В существующей версии портала реализована серия современных концепций, обеспечивающих построение распределенных web-систем, объединяющих на федеративной основе различные информационные источники – поставщики контента. Среди этих решений имеется и базовая концепция формирования пользовательского интерфейса, предполагающая разделение кода получения данных и кода представления.

Код представления реализуется в сайте портала на основе технологии портлетов. Этот код используется единственным и унифицированным внутренним механизмом портала для работы с данными, предоставляемым сервером интеграции и доступа. Механизм оперирует двумя понятиями: команды и набор данных. Под командой понимается XML-описание функции с параметрами, возвращающей набор данных (если команда на доступ), или признак завершения операции (если это команда модификации). В качестве набора данных выступает стандартная реляционная конструкция: набор таблиц с множеством связей. Этот набор данных представляет собой объект DataSet используемой платформы .Net.

Любые данные для кода представления поставляются сервером интеграции и доступа, обеспечивающим работу с адаптерами информационных источников. Если над данными выполнены некоторые действия, то эта же схема работает в другую сторону: измененный набор данных DataSet передается серверу интеграции и доступа, затем адаптеру, который и выполняет манипуляции с данными информационного источника. Таким образом, код получения данных в портале для любого подключенного источника должен представлять собой набор (простых) команд над наборами данных DataSet. Вызовы команд аналогичны вызову функции с параметрами.

Услуги, предоставляемые таким образом информационному источнику сервером интеграции и доступа и соответствующими адаптерами таковы, что позволяют достаточно многим существующим системам без каких-либо сложностей использовать их в качестве средства для подключения к portalу (см. рисунок).

Рассмотрим подробнее те отличия, которые имеет порталный адаптер в сравнении с традиционными адаптерами, выполняющими «аналогичную» функцию в электронных библиотеках.



Часть из них связана с необходимостью включения в адаптер дополнительной функциональности. Эта функциональность обеспечивает взаимодействие внешней системы не только с сервером интеграции и доступа, но и другими службами портала, в частности сайтом портала и подсистемой безопасности. Но есть и другое важное отличие. От традиционного адаптера требуется обеспечить только интерфейс к данным, обслуживаемым информационным источником. Таким образом, задача адаптера состоит в переводе запросов, сформулированных в терминах виртуальной (канонической) схемы репозитория (портала в данном случае), в запрос в терминах схемы источника. При этом, естественно, адаптер должен поддерживать язык запросов репозитория и обеспечивать преобразование не только схем, но и запросов на языке репозитория в запросы на языке источника, который в общем случае отличается (возможно, принципиально отличается, а то и отсутствует вовсе) от языка репозитория. В итоге задача создания адаптера может оказаться крайне сложной, сопоставимой, а то и более трудоемкой, с задачей создания собственно источника. В портале такая сложная схема не нужна. Предполагается, что можно ограничить взаимодействие портала и информационного источника конечным (не очень большим) перечнем запросов, каждому запросу поставить в соответствие строго одну функцию, и реализовывать в адаптере только ее. Благодаря этому предложению адаптер портала, а также сервер интеграции и доступа, да и вся технология взаимодействия с внешними системами, приобретают следующую специфику:

- функциональный интерфейс (отсутствует универсальный язык запросов, есть только функции);
- отсутствие, вообще говоря, схемы источника (есть только описание функций, схема данных источника может, при необходимости, быть извлечена из этих описаний); сама виртуальная схема репозитория портала представляет собой, по сути, набор описаний вызовов функций.

Конечно, в виртуальную схему портала входят и термины – описания хранимых объектов было бы нерационально каждый раз извлекать их из описаний функций. Однако, первичными остаются функции, на основании анализа которых уже строится виртуальная схема данных портала. При этом перестроение этой схемы происходит только в моменты подключения новых источников, к этому процессу привлекается администратор, большая часть действий которого поддерживается инструментарием, составляющим интерфейсную часть сервера интеграции и доступа.

Остается сказать какими именно внешними системами может пользоваться сервер интеграции и доступа, т.е. какие стандартные адаптеры есть в существующей реализации портала. Это практически любые хранилища структурированных данных, поддерживающие механизмы ADO, ODBC, SOAP, COM, DCOM. Другие (специальные) адаптеры довольно легко разрабатываются, для чего есть формальные требования, правила, рекомендации, в частности, описания программных интерфейсов, и множество примеров.

Любой источник, таким образом, подключается к portalу путем описания функций, выполняемых его адаптером в схеме сервера интеграции и доступа. Так, например, для реляционной базы данных, состоящий из десятков таблиц, содержащих описания двух-трех объектов предметной области, достаточно реализовать две-три функции (например, по одной на каждый объект, причем реализацией каждой такой функции будет команда Select, извлекающая нужный объект из базы в соответствии с заданными условиями-параметрами), добавить в схему сервера интеграции и доступа их описания, задать при необходимости атрибуты безопасности. В результате все интересные пользователю объекты этой базы данных могут быть показаны на страницах сайта портала. Причем изменять что-то в источнике не требуется. Кроме того, виртуальная схема портала автоматически расширится: в нее войдут описания тех объектов, которые предоставляются источником через описанные функции.

Наконец, несколько слов должны быть сказаны про платформу, использованную в существующей реализации. Это платформа Microsoft.Net, выбор которой был обусловлен следующими возможностями:

- наличие мощного класса DataSet, предназначенного для поддержки реляционных структур в памяти, и набора визуальных компонентов для работы с ним;
- построенная на .Net технология ASP.Net позволяет быстро и с минимальными затратами создавать web-приложения и web-интерфейсы пользователя;
- наличие готовой базовой инфраструктуры для построения систем безопасности;
- хорошая поддержка XML и web-сервисов.

#### 4 Реализация сервера доступа и интеграции

Из сказанного выше вытекает следующая схема работы портала (см. рисунок выше).

1. Любой код портала (в частности, при обслуживании пользовательских вызовов – код шаблона страницы сайта портала) работает с данными, используя набор данных, находящихся в экземпляре класса DataSet.

2. Для получения этого набора данных соответствующий код обращается к серверу интеграции и доступа, передавая ему имя команды (функции) и параметры для ее выполнения. При необходимости выполнения нескольких команд код может обратиться к серверу интеграции и доступа несколько раз или передать командный запрос – последовательность нескольких команд – и параметры для их выполнения. Так, например, сайт портала поддерживает собственные конфигурационные параметры, ставящие в соответствие каждому URL определенный командный запрос. Другой код может непосредственно содержать последовательность вызовов необходимых команд.

3. Для каждой вызванной команды сервер интеграции и доступа, пользуясь виртуальной схемой репозитория портала, определяет набор вызовов функций конкретных адаптеров, а также из параметров команды формирует параметры для вызова этих функций, затем последовательно вызывает эти функции.

4. Код реализации адаптера выполняет функцию (в простейшем случае просто перенаправляет вызов источнику, если тот сам поддерживает нужный программный интерфейс, в более сложном – сам работает с данными источника).

5. Результаты работы всех адаптеров возвращаются серверу интеграции и доступа, объединяются в один набор DataSet в соответствии с правилами объединения, описанными в схеме, и направляются осуществившей вызов исходной команды службе.

Сервер интеграции и доступа, таким образом, обеспечивает возможность работы с множеством подключенных информационных источников, пре-

доставляющих portalу структурированный контент, как с единым хранилищем, представленным в виде набора реляционных таблиц. Другими словами, любой код в порталных приложениях, осуществляющий доступ к данным с помощью сервера интеграции и доступа, «видит» все данные из подключенных информационных источников как единое множество реляционных таблиц. При этом сервер интеграции и доступа обеспечивает сопровождение общей (в рамках портала) виртуальной (канонической) схемы данных, объединяющей подсхемы всех информационных источников, подключенных к portalу. При этом заинтересованный в работе с данными код оперирует виртуальными терминами схемы портала, которые отображаются на термины конкретных информационных источников сервером интеграции и доступа.

Логически сервер интеграции и доступа представляется двумя частями: *схемой данных*, являющейся одновременно виртуальной схемой репозитория портала, и *командного процессора*, выполняющего все необходимые действия над командами. Схема данных представляет собой XML-файл (см. пример далее), соответствующий некоторой XSD-схеме, командный процессор реализован в виде программного модуля, представляющего собой .Net библиотеку.

Командный процессор, используя схему, выполняет преобразование конкретной команды из общей схемы в вызовы функций к соответствующим источникам, передает эти вызовы для выполнения соответствующим адаптерам, а затем преобразует (объединяет, сортирует, переименовывает) возвращенные результаты. Существуют как стандартные для портала адаптеры, входящие в существующую инфраструктуру, так и уникальные адаптеры, разработанные для ряда специализированных систем. Таким образом, источники структурированной информации, подключаемые к portalу через сервер интеграции и доступа, могут быть либо одного из predetermined типов, либо некоторого специального типа. Каждый predetermined тип источника уже имеет соответствующий адаптер, преобразующий вызовы функций в соответствующие вызовы для данного типа источника. В настоящий момент таких адаптеров три:

- XSServer (сервер приложений, обеспечивающий сервис внутреннего хранилища портала (его подробное рассмотрение выходит за рамки данной работы));
- для ADO.NET-источников;
- для SOAP-источников.

Подключение специального типа источника требует разработки специализированного адаптера, предназначенного для подключения именно этого типа информационного источника к portalу. В настоящий момент есть серия таких источников, из которых можно выделить архивы выдающихся ученых [5] и Интегрированную систему информационных ресурсов РАН [6,7]. Требования со стороны портала к разработке адаптеров для таких источни-

ков минимальны и обсуждаются далее. В любом случае собственно реализовывать в адаптере надо только те функции, которые предназначены владельцем источника для интеграции в портал.

#### 4.1 Команды манипулирования данными

Как уже упоминалось выше, взаимодействие с сервером интеграции и доступа осуществляется с помощью *команд манипулирования данными* (КМД). Каждая КМД фактически является XML-описанием функции с фиксированным списком параметров, возвращающей набор из одной или нескольких реляционных таблиц. Схема репозитория портала содержит описание всех КМД, которые может выполнить командный процессор.

Описание любой КМД в схеме содержит следующие элементы:

- уникальное имя КМД;
- список параметров КМД, с указанием имени, типа и обязательности каждого параметра;
- описание возвращаемого значения, если оно есть (в этом описании указывается количество возвращаемых таблиц и соответствующая им подсхема (набор таблиц и связей) портала);
- описание выполнения команды (подробнее см. далее).

КМД может модифицировать данные в репозитории источника, получать данные из репозитория или делать и то и другое. Причем выполнение одной КМД не обязательно будет модифицировать или получать данные только одной таблицы, одна КМД может манипулировать данными сразу в нескольких таблицах.

Программное взаимодействие с сервером интеграции и доступа обеспечивается единственной функцией – «Execute». Данная функция содержится в модуле (сборке .Net) XSUServer.dll, ее определение на языке C# выглядит следующим образом:

```
namespace XSUServer
{
    class XSUtils
    {
        static DataTable[]
Execute(string Command);
    }
}
```

Для выполнения последовательности КМД необходимо вызвать эту функцию, передав в параметре Command командный запрос – текст XML, описывающий последовательность вызовов КМД. Структура данного XML следующая:

```
<ExecuteBatch>
    <Имя_КМД имя_параметра="значение"
... />
...
</ExecuteBatch>
```

Типы параметров и формат передаваемых значений соответствуют спецификации [8]. Функция Execute последовательно выполняет КМД в порядке их перечисления в командном запросе. Если выполняемая КМД имеет возвращаемый результат, то он добавляется в конец массива результатов. После выполнения последней КМД, функция Execute возвращает полученный набор данных DataSet.

#### 4.2 Выполнение КМД

Из сказанного выше следует, что для выполнения КМД сервер интеграции и доступа пользуется функциями, предоставляемыми подключенными к нему адаптерами. Таким образом, реализация адаптера должна представлять из себя сборку .Net. Определение функций такой сборки на языке C# должно выглядеть следующим образом:

```
DataTable[] FunctionName(параметры );
```

Удовлетворяющие данному описанию функции могут иметь произвольное количество параметров и возвращать (возможно, пустой) массив объектов DataTable.

Сервер интеграции и доступа выполняет КМД в соответствии с *описанием выполнения*, содержащимся в схеме. Описание выполнения может быть либо кодом на языке .Net (C#, VB и т.п.), либо описанием последовательности вызова функций адаптеров.

Описание последовательности вызовов функций представляет собой последовательность блоков, каждый из которых описывает вызов одной функции определенного адаптера. В каждом таком блоке содержится следующая информация:

- имя сборки .Net и имя функции в этой сборке, которую необходимо вызвать;
- для каждого параметра вызываемой функции – выражение для вычисления значения этого параметра. В этом выражении могут использоваться имена параметров описываемой КМД, т.е. эта часть содержит правила преобразования параметров КМД в параметры конкретной функции адаптера;
- для каждой возвращаемой таблицы в массиве результатов (предполагается, что составитель описания знает количество и формат возвращаемых таблиц) правила помещения данных из этой таблицы в одну или несколько результирующих таблиц описываемой КМД. Эти правила по своим возможностям аналогичны оператору INSERT ... FROM SELECT ... языка SQL без соединений (JOIN) и группировки (GROUP BY), т.е. дает возможность добавить строки из одной таблицы в другую с фильтрацией строк по условию и переименованием, заменой или генерацией значений отдельных полей.

### 4.3 Схема репозитория

Сервер интеграции и доступа осуществляет свою деятельность на основе информационной структуры – *виртуальной схемы репозитория*. Схема представляет собой XML-файл, соответствующая XSD-Schema доступна по ссылке <http://www.ipiran.ru/XMLSchemas/SourceSchema.3.0.xsd>.

Виртуальная схема описывает множество реляционных таблиц и множество КМД, доступных посредством сервера доступа и интеграции. Про каждую таблицу в схеме содержится следующая информация:

- структура таблицы – описание имен и типов полей таблицы;
- индексы – первичный ключ и уникальные ключи;
- связи с другими таблицами – описание взаимосвязей таблиц в стиле «SQL foreign key».

Фактически такое описание набора таблиц аналогично традиционному описанию базы данных SQL.

Элементы описания в схеме (типы данных, ключей и т.п.) выбраны таким образом, что на основе схемы легко формируется структура таблиц в объекте .Net DataSet, и все элементы схемы отображаются на соответствующие элементы структуры данных объекта DataSet. Таким образом, использование сервера интеграции и доступа в среде .Net становится очень удобным. Кроме того, большинство распространенных источников данных (известные СУБД, OLE DB источники, некоторые XML источники) легко представляются с помощью объекта DataSet (именно с этой целью он и был разработан), а значит, и с помощью виртуальной схемы репозитория портала.

Как уже упоминалось выше, описание КМД в схеме содержит имя КМД, описание выполнения команды, описание параметров и возвращаемых значений. Описания КМД связаны с описаниями таблиц через описание возвращаемых результатов: если КМД возвращает какой-либо результат, то это должна быть одна или несколько описанных в схеме таблиц. Таким образом, в схеме данных можно считать первичным (основным) как описание таблиц, так и описание КМД. Если считать первичным описание таблиц, т.е. описание данных, тогда можно рассматривать всю схему как описание множества таблиц с присоединенным к каждой набором КМД. Если же считать первичным описание КМД, тогда схему можно рассматривать как описание доступных команд и описание данных, которыми эти команды манипулируют – возвращают, модифицируют и т.п.

Создание и поддержка виртуальной схемы – задача администратора портала. Именно он должен решать, как именно разные источники с разными функциями отображаются на схему репозитория портала. Простейшим примером является ситуация, когда схема просто содержит для каждой функции

каждого информационного источника отдельную КМД с собственным описанием возвращаемых результатов, т.е. никакие инструменты интеграции попросту не задействуются. Другой, «крайний» пример, когда создается некоторая всеобъемлющая схема данных, а затем составляется набор КМД, корректно работающих с этими данными, т.е. выполняющими запросы (возможно, к нескольким информационным источникам), осуществляющими преобразование структуры возвращаемых данных к схеме репозитория портала и т.д.

### 4.4 Адаптеры портала

При подключении нового источника к существующей схеме администратору необходимо сделать выбор: либо добавлять к существующей виртуальной схеме новые таблицы и КМД, либо встраивать вызовы функций вновь подключаемого информационного источника к существующим КМД, описывая, при необходимости, правила преобразования параметров и результатов. Решение по каждому вновь подключаемому источнику является индивидуальным и в значительной степени определяется семантикой подключаемого ресурса. В любом случае, внесенные в схему изменения должны поддерживаться указанной в качестве нового адаптера сборке .Net.

Для нового информационного источника может быть создан новый адаптер или может быть использован один из типовых. Существуют три типовых адаптера – XSServer (внутреннее хранилище портала), адаптер для доступа к ADO.Net источникам (на основе реляционных СУБД) и адаптер для доступа к SOAP источникам (web-сервисы), которые могут быть использованы при подключении соответствующих типов информационных источников.

XSServer представляет собой развитый сервер приложений. Он позволяет работать со специально подготовленными реляционными базами данных, поддерживает развитые средства управления доступом (авторизацией), объектное представление данных, наследование и др. Подробное описание этого сервера выходит за рамки данной работы.

Адаптер для доступа к ADO.Net источникам позволяет в качестве функции определить вызов хранимой процедуры или выполнение последовательности операторов языка SQL (SQL Batch) для определенного реляционного источника.

Адаптер для доступа к SOAP источникам позволяет вызывать функции указанного web-сервиса, осуществляя при этом преобразование результатов вызова функции web-сервиса к массиву объектов DataTable по заданным правилам.

### 4.5 Аутентификация и авторизация

Для осуществления аутентификации пользователей сервер интеграции и доступа взаимодействует с подсистемой безопасности портала, в частности с сервисом аутентификации. Идентификационные данные пользователя, выполняющего текущий за-



прос, доступны в выражениях, используемых в качестве параметров вызова функций адаптера с помощью специальных функций, предоставляемых API подсистемы безопасности портала. Таким образом, идентификационные данные пользователя, выполняющего запрос, могут быть переданы адаптеру (и, при необходимости, в информационный источник) для выполнения авторизации запроса.

Типовые адаптеры так же взаимодействуют со службой аутентификацией портала. Например, адаптер для SOAP включает идентификационные данные пользователя в вызов функции web-сервиса в соответствии со спецификацией WS-Security [9].

В случае, когда информационный источник сам не осуществляет авторизации доступа, а осуществлять ее необходимо, сервер интеграции и доступа предоставляет простой ролевой механизм авторизации на уровне вызова КМД: в схеме репозитория для любой КМД можно указать список ролей, которые должен иметь пользователь для вызова данной КМД.

## 5 Заключение

Представленная технология может быть названа технологией *функциональной интеграции*. Любой подключающийся к portalу источник может, используя сформулированные требования, проделать следующие шаги:

1. Описать функциональный интерфейс к своим данным с учетом накладываемых сервером интеграции и доступа ограничений.

2. Выбрать адаптер портала: если для данного типа источника есть готовый адаптер, то взять его, если готового адаптера нет, то его надо разработать.

3. Для стандартного адаптера описать свой функциональный интерфейс в терминах этого адаптера. Если адаптер нестандартный, то надо создать код реализации, используя один из языков .Net.

4. Решить каким образом и будут ли данные, возвращаемые созданным адаптером, интегрироваться с данными других информационных источников. Если нет, то функциональный интерфейс адаптера просто добавляется к виртуальной схеме. Если да, то для части функций (или для всех) уже есть аналоги в схеме, поэтому эти функции должны быть объединены с существующими функциями (с существующей виртуальной схемой), в т.ч. должны быть описаны правила объединения результатов в блоке описания выполнения каждой КМД.

5. Подключить источник к серверу интеграции и доступа, используя имеющийся административный интерфейс.

В итоге внешний информационный источник будет обеспечен следующим сервисом: данные и операции с информацией источника (в рамках функционально интегрированной части, т.е. в соответствии с перечнем реализованных в адаптере команд) будут доступны в едином стиле и общей структуре представления Информационного web-портала.

## Литература

- [1] Соколов И.А., Босов А.В., Бездушный А.Н. О Информационном Web-портале Российской академии наук // Системы и средства информатики. Вып. 13. – М.: Наука, 2003.
- [2] Босов А.В., Иванов А.В. О реализации системы управления содержанием информационного Web-портала // Информационные технологии и вычислительные системы. №4. – М.: ИМВС РАН, 2004.
- [3] Россия. Веб-портал РАН обеспечивает эффективный доступ научного сообщества к актуальной информации. Информационный бюллетень Microsoft. Выпуск № 26. Ноябрь 2004 года.
- [4] Глеб Галкин. Такие разные корпоративные порталы // «Сетевой журнал», №5, 2002. <http://www.setevoi.ru/cgi-bin/text.pl/magazines/2002/5/54>.
- [5] Босов А.В., Чавтараев Р.Б. Научное наследие: личные архивы выдающихся ученых в портале РАН // в печати, 2006.
- [6] Бездушный А.А., Бездушный А.Н., Нестеренко А.К., Серебряков В.А., Сысоев Т.М. Предложения по наборам метаданных для научных информационных ресурсов ЕНИП РАН // Электронный журнал, посвященный созданию и использованию электронных библиотек, том 7, выпуск 5. Москва: Институт развития информационного общества - 2004. <http://www.elbib.ru/index.phtml?page=elbib/rus/journal/2004/part5/bnss>.
- [7] Бездушный А.А., Нестеренко А.К., Сысоев Т.М., Бездушный А.Н., Серебряков В.А. Возможности технологий ИСИР в поддержке Единого Научного Информационного Пространства РАН // Электронный журнал, посвященный созданию и использованию электронных библиотек, том 6, выпуск 4. Москва: Институт развития информационного общества - 2004. <http://www.elbib.ru/index.phtml?page=elbib/rus/journal/2004/part6/bnsbs>.
- [8] XML Schema Part 2: DataTypes, Primitive DataTypes // <http://www.w3.org/TR/xmlschema-2/>.
- [9] Web Services Security // <http://www-106.ibm.com/developerworks/webservices/library/ws-secure/>.

## On Data Access Technology in Informational Web-Portal

A.V.Bosov, A.N.Poluhin

The solution including technology and realization used in Informational web-portal for different external information sources content support is considered. Proposed integration and access server architecture is based on the external systems functional integration concept and adapters technology evolution.