

УДК 004.65.3

Росинский В. В., научный сотрудник

ИНТЕГРАЦИЯ ОБЪЕКТОВ БАЗ ДАННЫХ MS SQL SERVER 2000 В КОРПОРАТИВНЫЕ ИНФОРМАЦИОННЫЕ СИСТЕМЫ НА ОСНОВЕ ТЕХНОЛОГИИ ADOBE FLEX

Росінський В.В. Інтеграція об'єктів баз даних MS SQL Server 2000 в корпоративні інформаційні системи на основі технології Adobe Flex. Розглядаються переваги використання сервіс-орієнтованої архітектури в корпоративних інформаційних системах. Пропонується найбільш зручний спосіб інтеграції об'єктів баз даних з клієнтськими програмами з використанням новітніх технологій web-інтеграції на основі web-сервісів. Демонструється алгоритм створення web-сервісу на основі SQL Server 2000 і програми в середовищі Adobe Flex, що є клієнтом служби.

Ключові слова: КОРПОРАТИВНІ ІНФОРМАЦІЙНІ СИСТЕМИ, WEB-ІНТЕГРАЦІЯ, WEB-СЕРВІСИ, WSDL-ОПИС, ADOBE FLEX, MS SQL SERVER, SOA, SOAP, XML

Росинский В. В. Интеграция объектов баз данных MS SQL Server 2000 в корпоративные информационные системы на основе технологии Adobe Flex. Рассматриваются преимущества применения сервис-ориентированной архитектуры в корпоративных информационных системах. Предлагается наиболее удобный способ интеграции объектов баз данных с клиентскими приложениями с использованием новейших технологий Web-интеграции на основе Web-сервисов. Демонстрируется алгоритм создания Web-службы на основе SQL Server 2000 и приложения-клиента в среде Adobe Flex, которое использует службу.

Ключевые слова: КОРПОРАТИВНЫЕ ИНФОРМАЦИОННЫЕ СИСТЕМЫ, WEB-ИНТЕГРАЦИЯ, WEB-СЕРВИСЫ, WSDL-ОПИСАНИЕ, ADOBE FLEX, MS SQL SERVER, SOA, SOAP, XML

Rosinskyi V. V. Integration of MS SQL Server 2000 databases objects in the corporate information systems on the basis of Adobe Flex technology. Advantages of application of service-oriented architecture in corporate information systems are examined. The most comfortable method for integration of objects of databases with client applications using the newest technologies of web-integration on the basis of web-services is proposed. The algorithm of creation of web-service is demonstrated on the basis of SQL Server 2000 and client application in Adobe Flex environment.

Keywords: CORPORATE INFORMATION SYSTEMS, WEB-INTEGRATION, WEB-SERVICES, WSDL-DESCRIPTION, ADOBE FLEX, MS SQL SERVER, SOA, SOAP, XML

Введение. Современные задачи корпоративных информационных систем (КИС), учитывая тенденцию к консолидации, слияниям и поглощениям, требуют гибкости в управлении бизнес-процессами и соответствующей адаптации ИТ-инфраструктуры. Одна из насущных и сложных проблем ИТ-отрасли – это интеграция. Если на первоначальном этапе внедрения КИС стояла задача создания информационной системы “с нуля”, то сегодня требуется оптимизация и доработка уже сложившейся ИТ-инфраструктуры. Неизбежно возникают проблемы, обусловленные разрозненностью используемых в отдельных структурных подразделениях локальных систем. Поэтому и встает задача их интеграции. С уверенностью можно сказать, что сейчас практически любой ИТ-проект предполагает проведение ряда работ, связанных с интеграцией приложений, инструментальных средств или программных продуктов. Доля их постоянно увеличивается. Вследствие этого возникла потребность в выделении данной проблемы как такой что требует фундаментального подхода. Информация, необходимая для эффективного принятия решений по управлению, содержится во множестве разрозненных источников организации (базах данных, операционных системах, бизнес-приложениях и т. п.).

Особо следует отметить тот факт, что всегда присутствует стремление сохранить инвестиции, сделанные в ИТ. Технологии разработки программного обеспечения достигли такого уровня, что позволяют использовать уже ставшие привычными и удобными для пользователей корпоративные приложения путем объединения их на базе единой интеграционной платформы [1].

Построение КИС на базе SOA. Развитие системной интеграции, опирающееся на новые передовые технологии, стандарты и положительный опыт ИТ-индустрии, повлекло за собой создание сервис-ориентированной архитектуры (Service-Oriented Architecture, SOA). SOA

представляет собой общий подход к построению архитектуры КИС, при котором функциональные элементы приложений представляются как совокупность взаимодействующих Web-сервисов, которые могут быть использованы другими приложениями. Web-сервисы – это доступные через локальную или глобальную сеть автономные программные модули, обладающие максимально стандартизованным интерфейсом и предназначенные для выполнения конкретных функций обслуживания пользователей и обеспечения бизнес-процессов. Они реализуют широкий спектр функций и могут легко комбинироваться и повторно использоваться для выполнения различных бизнес-задач. Наибольшую ценность SOA представляет для компаний, обладающих несколькими разнородными системами, автоматизирующими управление в рамках функциональных направлений деятельности (компании с распределенной гетерогенной IT-инфраструктурой). В настоящее время концепция SOA признана во всем мире как наиболее передовой подход к созданию целостной информационной среды, позволяющий существенно сократить время и стоимость реализации проектов. SOA не зависит от языков программирования, платформ или протокольных спецификаций.

Однако есть мнение что SOA – это термин, который появился для описания исполняемых компонентов, таких как Web-сервисы, которые могут вызываться другими программами, выступающими в качестве клиентов или потребителей этих сервисов. Эти сервисы могут быть полностью современными или устаревшими прикладными программами, которые можно активизировать как черный ящик. От разработчиков не требуется знать, как работает Web-сервис, необходимо лишь понимать, какие входные и выходные данные необходимы, и как вызывать Web-сервис для исполнения.

В самом общем виде SOA предполагает наличие трех основных участников: поставщика сервиса, потребителя сервиса и реестра сервисов (рис. 1). Взаимодействие участников выглядит следующим образом: поставщик сервиса регистрирует свои сервисы в реестре, а потребитель обращается к реестру с запросом.



Рис. 1. Общая схема SOA

Для использования Web-сервиса необходимо следовать соглашению об интерфейсе для обращения к сервису – интерфейс должен не зависеть от платформы. SOA реализует масштабируемость сервисов – возможность добавления сервисов, а также их модернизацию. Поставщик сервиса и его потребитель оказываются несвязанными – они общаются с помощью сообщений. Поскольку интерфейс должен не зависеть от платформы, то и технология, используемая для определения сообщений, также должна не зависеть от

платформи. Поэтому сообщения являются XML-документами, которые соответствуют XML-схеме.

Web-сервисы составляют одно из наиболее значимых преимуществ технологии Microsoft.NET. На сегодняшний день огромное количество разработчиков программного обеспечения приспособились к распределенным средам. Это не только обеспечивает их проектам масштабируемость и доступность, но также и значительно уменьшает время разработки, вследствие увеличенной возможности многократного использования программного кода. В прошлом сеанс клиент-сервер часто ограничивался и операционной системой и платформой разработки, так же как протоколами связи. С Web-сервисами эти ограничения были сняты, за счет открытых стандартов, таких как XML и HTTP.

HTTP был выбран как транспортный протокол для Web-сервисов вследствие его универсального характера. Его использование позволило передавать данные через корпоративные брандмауэры, что сделало возможным совместное использование Web-сервисов вне сред интранета. Наряду с доступностью использования запросов HTTP GET и HTTP POST, обмен сообщениями между клиентом и Web-сервисом осуществляется через SOAP (Simple Object Access Protocol). Сообщения SOAP это форматированные XML-документы, в которых указывается схема, определяющая их формат, и содержится ссылка на метод, который будет выполняться, также его параметры или результаты (в зависимости от того, является ли сообщение запросом SOAP или ответом SOAP). XML также используется в качестве языка описания Web-сервисов (WSDL) для их публикации, что позволяет клиентам определять методы доступны в Web-сервисе, а также параметры, требующиеся для выполнения этих методов [2].

Открытые стандарты, описывающие XML и Web-сервисы, позволяют применять SOA ко всем технологиям и приложениям, задействованным в КИС. Как известно, Web-сервисы базируются на широко распространенных и открытых протоколах: HTTP, XML, UDDI, WSDL и SOAP. Именно эти стандарты реализуют основные требования SOA к Web-сервисам:

- поддаются динамическому обнаружению и вызову благодаря UDDI, WSDL и SOAP;
- используют независимый от платформы интерфейс на основе XML.
- обеспечивают функциональную совместимость за счет протокола HTTP [3].

Основным достоинством Web-сервисов является их платформенезависимость. Предприятия могут работать с Web-сервисами как на огромных универсальных ЭВМ фирмы Hitachi, так и на небольших серверах Microsoft Windows NT, а используемое программное обеспечение может быть собственным или лицензионным.

Не так давно была описана трехуровневая архитектура Web-сервисов. Ее создали Джон Сили Браун (John Seely Brown), ведущий научный сотрудник компании Xerox, и Джон Хагел III (John Hagel III), главный аналитик компании 12 Entrepreneuring, расположенной в Сан-Франциско, Калифорния. На нижнем уровне были стандарты и протоколы, такие как XML, SOAP и TCP/IP. Второй уровень они назвали "сетка сервисов". Аналогично электрической сети, она состоит из общих утилит, которые составляют, например, счета третьих фирм, сервисы безопасности и обмена сообщениями, которые являются общими и могут использоваться многими приложениями. На верхнем уровне этой иерархии Хагеля-Брауна находятся собственно прикладные сервисы. Они, по определению, являются специфическими для каждой конкретной КИС [4].

Корпорация Microsoft выпустила для SQL Server 2000 набор инструментов Web Services Toolkit, в состав которого входит SQLXML 3.0 – третья редакция средства построения XML-приложений для SQL Server 2000. Web Services Toolkit – это набор инструментов, позволяющий графически создавать Web-службы для таких объектов баз данных, как хранимые процедуры (stored procedures, SP) и определяемые пользователем функции (user-defined functions, UDF) [5]. С момента появления SQLXML 3.0. Microsoft выпустила уже два пакета обновлений для данного продукта. Последнее обновление, SP2, можно загрузить с сайта компании.

Постановка проблеми. Поскольку для фундаментальной науки характерно ограниченное финансирование, проблемы с разработкой и исследованием возможностей применения Web-сервисов в научных проектах решаются медленно. Более значимых успехов в глобальной стандартизации разработки и использования Web-сервисов добились коммерческие организации, которые могут позволить себе инвестировать в это перспективное направление большие суммы денег [6]. В литературе посвященной данной тематике проводится поверхностное исследование и описание применения отдельных инструментов интеграции SQL Server 2000 и клиентских приложений использующих Web-сервисы. На данный момент в научной среде отсутствует комплексный подход, и глубокий анализ применения имеющихся в наличии возможностей интеграции объектов SQL Server 2000 в КИС.

Цель данной статьи – провести исследование возможностей интеграции SQL Server 2000 с клиентскими приложениями на основе Web-сервисов в среде Flex и продемонстрировать комплексный алгоритм создания гибкой Web-службы и приложения-потребителя, вследствие которого расширяет возможности использования существующих объектов базы данных.

Виртуальный каталог Web-сервиса. Исследование интеграционных возможностей сервера баз данных SQL Server 2000 следует начать с создания виртуального каталога Web-службы, который является промежуточным уровнем между уровнем базы данных и уровнем пользователя. Этот уровень можно рассматривать как отображение интерфейса Web-сервиса на базу данных SQL Server 2000. Поскольку есть возможность передачи параметров Web-методам среднего уровня, которые выполняют различные действия типа изменения строки в таблице, то виртуальный каталог должен располагать сведениями о том, где находится база данных и какие конкретно объекты используются для выполнения действий. Виртуальный каталог должен также иметь возможность регистрироваться в базе данных, поскольку права учетной записи неявно определяют права и набор действий, которые может выполнять Web-сервис. Кроме того, виртуальный каталог должен реализовывать базовую функциональность SOAP и WSDL (Web Service Description Language, язык описания Web-сервисов), чтобы любой клиент мог задействовать стандартные средства обращения к Web-сервису.

Запуска программы управления виртуальными каталогами SQLXML 3.0 IIS Virtual Directory Management осуществляется через Меню→Пуск→SQLXML 3.0→Configure IIS Support. Этот инструмент позволяет создавать новые виртуальные каталоги или изменять настройки существующих каталогов. На рис. 2 изображена вкладка General диалогового окна, позволяющая создавать виртуальные каталоги, управлять ими и выполнять соответствующие настройки для рассматриваемого в данной статье Web-сервиса.

В поле Virtual Directory Name необходимо задать имя виртуального каталога – *DBSales*. В поле Local Path указывается путь к папке, в которой будут размещаться файлы доступные посредством виртуального каталога – *C:\Interpub\wwwroot\DBSales*.

Вкладка Security диалогового окна позволяет указать, каким образом пользователи Web-сервиса будут подключаться к уровню базы данных.

На вкладке Data Source в поле SQL Server необходимо выбрать сервер баз данных, объекты которых будут использоваться посредством виртуального каталога. В поле Database необходимо указать имя используемой базы данных – *DBSales*.

При настройке виртуального каталога для Web-сервиса на вкладке Settings необходимо установить флажок Allow Post на закладке Settings (по умолчанию отключен). Этот параметр позволяет приложению отправлять значения виртуальному каталогу и таким образом взаимодействовать с Web-сервисом. Все остальные флажки на вкладке Settings можно отключить.

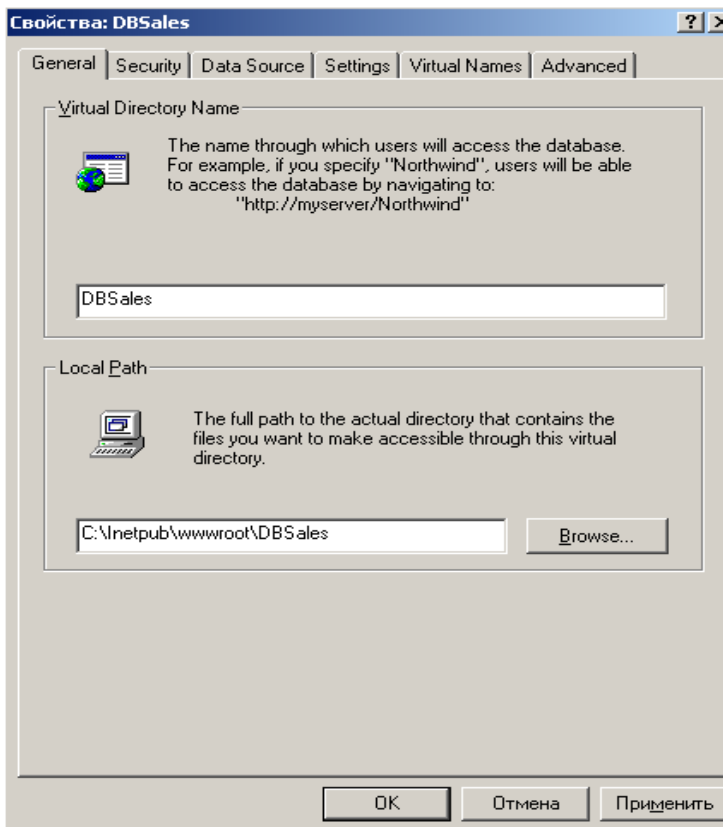


Рис. 2. Диалоговое окно SQLXML 3.0 IIS
Virtual Directory Management

На вкладке Virtual Names необходимо выбрать <New virtual name>. В списке Type нужно выбрать SOAP, чтобы показать, что виртуальный каталог содержит объект SOAP для описываемого Web-сервиса. При выборе объекта SOAP происходит автоматическое заполнение поля Domain name именем домена, указанным для виртуального каталога на вкладке General. При этом необходимо самостоятельно ввести имя Web-сервиса и путь к содержащей его папке. Для присвоения имени объекту SOAP используется текстовое поле Name. В текстовом поле Path следует ввести локальный путь для Web-сервиса - *C:\Interpub\wwwroot\DBSales*. Для передачи изменений, внесенных в объект SOAP в виртуальном каталоге необходимо нажать кнопку Save, а затем нажать Configure.

Настройка объекта SOAP в первую очередь означает определение объектов базы данных, которые будут обрабатываться Web-сервисом, и формат значений, возвращаемых клиентскому приложению. Для определения нового метода Web-сервиса необходимо выбрать <New method mappind>. Поскольку клиенту будет передаваться XML-документ, который является результатом работы хранимой процедуры, следует выбрать тип SP. Далее в поле SP/Template необходимо выбрать необходимую хранимую процедуру из списка, которая будет предоставлена как Web-метод. Поле Method name можно оставить без изменений.

Выходной формат для значений, возвращаемых в клиентское приложение, определяется выбранным значением «Output as» и, конечно, объектом базы данных, выбранным для реализации Web-сервиса. Хранимая процедура возвращает результирующий набор, который Web-сервис представляет как XML-документ. Для этого типа возвращаемых значений нужно выбрать переключатели XML objects.

После нажатия кнопки Save будет создан объект SOAP. После этого в каталоге *C:\Interpub\wwwroot\DBSales* будут созданы 2 файла. Один из них с расширением .ssc – служебный, второй с расширением .wsdl. Его содержание необходимо проанализировать для создания Flex-клиента.

Анализ WSDL-документа. В разделе <types> WSDL-документа определяются все используемые Web-сервисом типы данных, включая специальные типы данных и форматы сообщений. В нем следует отыскать часть, в которой описан необходимый метод Web-сервиса. Из нее можно узнать имя и названия параметров используемого метода. Имя Web-метода необходимо указать в компоненте <mx:WebService> клиента Flex, в секции <mx:operation> в параметре name. Названия параметров используются в секции <mx:operation> в теге <mx:request>. Здесь важно указать точное название и количество параметров метода.

Раздел <message> WSDL-документа содержит информацию о сообщениях запроса и ответа, используемых для обмена данными с Web-сервиса. Сообщения представляют информацию, которой метод Web-сервиса и клиент обмениваются между собой. Данные из этой секции использоваться во Flex-клиенте не будут.

Информация в разделе <portType> WSDL-документа представляет собой своего рода каталог доступных в данном Web-сервисе функциональных возможностей – детальной информация о предназначении каждого метода Web-сервиса. В отличие от раздела <message>, который содержал независимые элементы для входного и выходного сообщения, операции в этом разделе связаны вместе по схеме “запрос-ответ”. Имя операции – это имя метода.

Раздел <binding> содержит информацию о транспортных протоколах, которые поддерживает данный Web-сервис. Элементы этого раздела связывают абстрактный формат данных с конкретным протоколом, который используется для пересылки данных через Интернет соединение. С помощью элемента <binding> WSDL-документ описывает низкоуровневый протокол связи, который может использоваться для взаимодействия с Web-сервисом. Он связывает этот элемент с элементом <operation> из раздела <portType>.

В разделе <service> определяются входные точки Web-сервиса в виде одного или более элементов <port>. Каждый элемент <port> содержит информацию об адресе или URI. Тег <wsdl:service name...> содержит название Web-сервиса, которое будет использовано во Flex-клиенте в атрибуте *service* компонента <mx:WebService>. Тег <wsdl:port name...> содержит название порта. Его необходимо задать в атрибуте *port* компонента <mx:WebService> клиента Flex.

Построение клиента. В код клиента необходимо добавить тег <mx:WebService>, который отвечает за компонент WebService. В атрибуте *wsdl* задается URL-адрес документа WSDL с описанием Web-сервиса. Адрес формируется следующим образом:

http://Server_name/Directory_name/SOAPobject_name.wsdl

где: *Server_name* – имя сервера; *Directory_name* – имя директории где содержится WSDL-файл; *SOAPobject_name* – имя Web-сервиса.

Как уже отмечалось ранее в атрибуте *service* необходимо задать *service name* из раздела <service>, а в атрибуте *port* необходимо указать *port name* из раздела <service>:

Все используемые Web-сервисом методы описываются тегами <mx:Operation>. Их имена и описания находятся в разделе <portType>. Для упрощения дальнейшей обработки XML-документа полученного от Web-сервиса Flex-Клиентом, атрибут *resultFormat* необходимо изменить на "e4x". В теге <mx:request> указываются параметры используемого метода. Их название, количество и тип указаны в разделе <types> WSDL-документа.

Для отображения результатов работы хранимой процедуры Web-метода в код клиента добавляется компонент DataGrid, который представляет собой таблицу. В его свойстве *dataProvider* необходимо указать следующую строку:

IDWeb-Service.Method1.lastResult.Root1...RootN.Resultroot

где: *IDWeb-Service* – индификатор компонента <mx:WebService>; *Method1* – название вызываемого метода; *lastResult* – свойство компонента <mx:WebService>, в котором храниться результат работы данного метода; *Root1...RootN* – корневые элементы верхних уровней полученного XML-документа; *Resultroot* – узел XML-документа, в котором непосредственно храниться результат запроса.

На рис. 3 слева отображен результат работы хранимой процедуры запущенные посредством MS SQL Server 2000. Справа – результат работы той же самой хранимой процедуры, но уже как Web-сервиса, в окне браузера в компоненте DataGrid. Клиентом Web-сервиса в данном случае является браузер.

ProductName	Total
Boston Crab Meat	50
Camembert Pierrot	56
Chang	24
Escargots de Bourgogne	30
Flotemysost	120
Gorgonzola Telino	56
Gravad lax	28
Guaraná Fantástica	43
Gudbrandsdalsost	25
Gumbär Gummibärchen	24
Inlagd Sill	60
Jack's New England Clam Chowder	20
Lakkalikööri	15

Рис.3. Отображение результатов хранимой процедуры посредством MS SQL Server 2000 и браузера.

Выводы. В представленной статье продемонстрирован подход, при котором обеспечивается взаимодействие браузерного приложения в качестве клиента Web-сервиса размещенного на сервере MS SQL Server 2000. Созданное трехуровневое приложение на основе Web-служб, состоит из базы данных промежуточного уровня и уровня клиента, работающее как единое интегрированное приложение, но при этом, по сути, являющееся слабосвязанным. Инструментарий виртуального каталога SQLXML 3.0 IIS Virtual Directory Management создает средний уровень как виртуальный каталог, позволяющий обращаться к объектам базы данных. В Web-службе создан объект SOAP, представляющий объекты базы данных как Web-методы. Данный подход свидетельствует о том, что посредством интеграционных Web-технологий можно обеспечить кроссплатформенность и интероперабельность в корпоративных информационных системах, где в качестве источника данных используются сервера баз данных MS SQL Server 2000, с целью расширения их возможностей.

Литература

1. Анфиногентов В. А. ИТ-интеграция: современный подход и инструментальные средства / В. А. Анфиногентов // Connect! Мир Связи. – 2006. – №10. – С. 51-57.
2. Policht M. SQL 2000 and Web Services / Policht M. // Database Journal. – 2003. – №9. – Р. 34-40.
3. <http://www.information-management.com/issues/20050101/1016488-1.html>
4. Sandsmark F. Make Way for Web Services / Sandsmark F. // Oracle's E-Business Magazine. – 2002. – №2. – Р. 46-52.
5. Cerami E. Web Services Essentials. Distributed Applications with XML-RPC, SOAP, UDDI & WSDL / Cerami E. – O'Reilly Media, 2002. – P.121-127.
6. Исследование и разработка системы метаданных для электронных информационных ресурсов и сервисов в фундаментальной науке / А. Б. Антопольский, В. И. Ауссем, С. А. Блау, А. И. Жежель // Отчет о результатах работ по гранту РФФИ № 04-07-90087.