**Веб-сервисы**

Веб-сервисы меняют все! Точно так же, как железнодорожные перевозки влияют на экономику страны, так и веб-сервисы коренным образом изменяют правила электронной коммерции. Передавая большой объем данных более эффективно и дешевле, чем ранее, они обеспечивают связь между приложениями, выполняемыми в любых удаленных точках земного шара. В результате между предпринимателями и потребителями устанавливается быстрая, удобная и продуктивная связь.

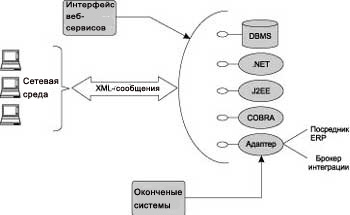
Сеть в ее современном состоянии плохо поддерживает программно-ориентированное взаимодействие. Частным лицам и компаниям, занимающимся коммерцией с использованием сети Интернет, необходим способ публикации ссылок на их приложения и данные, практически такой же, как при публикации ссылок на их веб-страницы. Интернет-приложения должны иметь возможность находить другие интернет-приложения, обращаться к ним и автоматически взаимодействовать с ними. Обеспечивая "общение" между программами, веб-сервисы расширяют возможности Интернета. За счет поддержки веб-сервисов размещенные в различных узлах Интернета приложения могут взаимодействовать непосредственно, как будто они являются частью одной крупной информационной системы.

**Основы веб-сервисов**

Веб-сервисы преобразуют XML-документы (Extensible Markup Language, XML) в ИТ-системах. Веб-сервисы - это XML-приложения, осуществляющие связывание данных с программами, объектами, базами данных либо с деловыми операциями целиком. Между веб-сервисом и программой осуществляется обмен XML-документами, оформленными в виде сообщений. Стандарты веб-сервисов определяют формат таких сообщений, интерфейс, которому передается сообщение, правила привязки содержания сообщения к реализующему сервис приложению и обратно, а также механизмы публикации и поиска интерфейсов.

Веб-сервисы могут использоваться во многих приложениях. Независимо от того, откуда запускаются веб-сервисы, с настольных компьютеров клиентов или с переносных, они могут использоваться для обращения к таким интернет-приложениям, как система предварительных заказов или контроля выполнения заказов. Веб-сервисы пригодны для В2В-интеграции (business-to-business), замыкая приложения, выполняемые различными организациями, в один производственный процесс. Веб-сервисы также могут решать более широкую проблему интеграции приложений предприятия (Enterprise Application Integration, EAI), осуществляя связь нескольких приложений одного предприятия с несколькими другими приложениями, размещенными как "до", так и "после" брандмауэра. Во всех перечисленных случаях технологии веб-сервисов являются "связующим звеном", объединяющим различные части программного обеспечения.

Как видно из рис. 1.1, веб-сервисы представляют собой оболочку, обеспечивающую стандартный способ взаимодействия с прикладными программными средами, такими как системы управления базами данных (СУБД), .NET, J2EE (Java2 Platform, Enterprise Edition), CORBA (Common Object Request Broker Architecture), посредники пакетов планирования ресурсов предприятия (Enterprise Resource Planning, ERP), брокеров интеграции и пр.



*Рис. 1.1. Веб-сервисы взаимодействуют с прикладными системами*

Интерфейсы веб-сервисов получают из сетевой среды стандартные XML-сообщения, преобразуют XML-данные в формат, "понимаемый" конкретной прикладной программной системой, и отправляют ответное сообщение (последнее - не обязательно). Программная реализация веб-сервисов (базовое программное обеспечение, нижний уровень) может быть создана на любом языке программирования с использованием любой операционной системы и любого связующего программного обеспечения (middleware).

Веб-сервисы объединяют программирование и концепции Сети. Веб-сервисы сочетают параметры программных приложений и абстрактные характеристики Сети. Современные интернет-технологии частично достигают своих целей, поскольку они определены на очень высоком отвлеченном уровне, что обеспечивает совместимость с любой операционной системой, любым программным и аппаратным обеспечением. Инфраструктура, основанная на применении веб-сервисов, пользуется этим уровнем абстракции и включает в себя связанную с данными семантическую информацию, то есть веб-сервисы определяют не только данные, но и порядок обработки и преобразования этих данных в базовые программные приложения и обратно.

**Простой пример: поиск информации**

В настоящее время большинство сервисов вызываются по Сети посредством ввода данных в HTML-формы и отправки этих данных сервису путем добавления их в строку унифицированного указателя информационного ресурса (Uniform Resource Locator, URL):

<http://www.google.com/search?q=Skate+boots&btnG=Google+Search>

Этот пример иллюстрирует простоту веб-взаимодействия (например, поиска, покупки акций или запроса маршрута движения), где параметры и ключевые слова внедряются непосредственно в URL. В данном случае представлен простой запрос поиска skate boots (ботинки с коньками) в строке обращения к поисковой машине Google. Ключевое слово search (искать) представляет сервис, к которому будет осуществлено обращение, а параметр Skate+boots является строкой поиска, которая была введена в HTML-форме на странице веб-сайта Google. Сервис поиска Google передаст этот запрос к различным поисковым машинам, которые вернут список URL для страниц, на которых имеется соответствие параметру поиска Skate+boots. Данный малоэффективный способ поиска в Сети полностью основан на установлении соответствия указанной текстовой строки и индексированных HTML-страниц.

XML - лучший способ отправки данных. XML предоставляет значительные преимущества при передаче данных через Интернет. Теперь предыдущий запрос можно представить в виде XML-документа:

<SOAP-ENV:Body>

<s:SearchRequest

xmlns:s="www.xmlbus.com/SearchService">

<pl>Skate</pl>

<p2>boots</p2>

<p3>size 7.5</p3>

</s:SearchRequest>

</SOAP-ENV:Body>

Отправка запроса в виде XML-документа имеет следующие преимущества: возможность определения типов данных и структур, большую гибкость и расширяемость. XML может представлять структурированные данные или данные определенного типа (например, допустимо указывать значение поля size (размер) как в виде строки цифр, так и в форме числа с плавающей точкой) и содержать больший объем информации, чем это допускает URL.

Данный пример представлен в форме SOAP-сообщения (Simple Object Access Protocol) - стандартной формы обмена XML-сообщениями - одной из технологий, лежащих в основе веб-сервисов. В SOAP-сообщении имя запрашиваемого сервиса и входные параметры представлены в виде отдельных XML-элементов. Рассматриваемый пример также иллюстрирует использование пространства имен XML (xmlns:), еще одного важного элемента веб-сервисов. Благодаря тому, что XML-документы поддерживают разные типы данных, сложные структуры и объединение схем, современные технологии веб-сервисов обеспечивают значительное преимущество над существующими возможностями обращения к программным приложениям посредством HTML и URL.

**Следующее поколение Сети**

Следующее поколение Сети будет основано на программно-ориентированных взаимодействиях. Веб-сервисы предполагают использовать созданные для взаимодействия людей глобальные сети совершенно в иных целях. Программно-ориентированные взаимодействия будут автоматически выполнять операции, которые ранее обязательно требовали "ручного" вмешательства:

 поиск и покупка товаров и услуг по самой выгодной цене;

 согласование заказов авиабилетов и мест в ресторане на определенную дату (планирование путешествий);

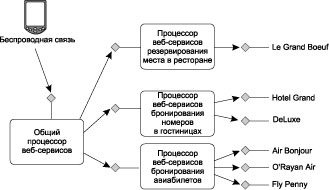
 оптимизация коммерческих операций закупки товаров, выписки счетов и доставки.

Следующее поколение Сети будет использовать программно-ориентированные сервисы для непосредственного взаимодействия с приложениями, построенными на основе любой комбинации объектов, программ и баз данных.

Веб-сервисы являются не только интерфейсом объектов, программ, связующего программного обеспечения и баз данных для доступа по Сети. Объединение ряда веб-сервисов позволяет осуществлять новые типы взаимодействия.

Веб-сервисы предоставляют новые варианты взаимодействия. Представим, например, что вы живете в Сан-Франциско и желаете зарезервировать столик в любимом парижском ресторане, а затем составить план путешествия, позволяющий оказаться там в необходимое время. В настоящее время для осуществления заказа необходимо непосредственно позвонить в ресторан, учитывая 9-часовую разницу во времени и различия в языке, а затем связаться с агентом бюро путешествий и подобрать подходящий рейс и гостиницу. Однако, используя веб-сервисы, можно спланировать обед с помощью календаря карманного компьютера (Personal Digital Assistant, PDA), а для автоматического резервирования столика на определенное время просто щелкнуть на кнопке. После выполнения резервирования веб-сервис запустит другие сервисы, которые забронируют наиболее дешевые авиабилеты и закажут номер в самом ближайшем четырехзвездочном отеле.

На рис. 1.2 представлен порядок осуществления взаимодействия веб-сервисов с карманным компьютером, подключенным по беспроводному каналу к процессору веб-сервисов, который для заказа столика в любимом ресторане использует веб-сервис этого ресторана. Процессор веб-сервисов получает запросы от функции "календарь" карманного компьютера и находит веб-сервисы, связанные с расширенной функцией календаря, такой как заказ столика. Для завершения планирования путешествия после успешного резервирования места в ресторане процессор веб-сервисов контактирует с веб-сервисами бронирования авиабилетов и номеров в гостиницах.

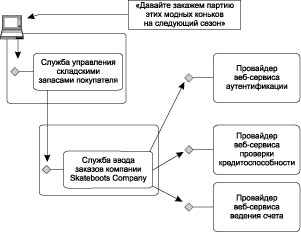


*Рис. 1.2. Приложения могут использовать веб-сервисы для резервирования места в ресторане   
и бронирования авиабилетов и номеров в гостиницах*

Веб-сервисы способны находить другие сервисы и взаимодействовать с ними. Веб-сервисы также полезны при поиске и осуществлении взаимодействия с сайтами, являющимися интерактивными системами заказов, например с системой продаж модных коньковых ботинок с выдвигающимися лезвиями.

Розничные продавцы спортивных товаров, заинтересованные в создании запасов таких ботинок, могут использовать веб-сервисы для размещения предварительных заказов, для проверки состояния выполнения заказа или размещения заказов сезонного возобновления запасов. Они также получат немедленный ответ при отсутствии коньков на складе. "Строительные" блоки веб-сервиса предоставляют стандартные компоненты конкретного приложения для компании Skateboots Company, которая не настолько велика, чтобы содержать свою собственную инфраструктуру приложений. Компания-владелец веб-сервиса также обеспечивает безопасность, гарантируя получение заказов компанией Skateboots Company только от уполномоченных оптовых покупателей, и предлагает сервисы проверки полномочий для крупных предварительных заказов. Другие компании помогут Skateboots Company создать электронный каталог и выполнить расчеты.

Целиком реализованная как веб-сервис система ввода заказов компании Skateboots зависит от Интернета, но необходимая функциональность обеспечивается за счет совместной работы ряда других веб-сервисов. На рис. 1.3 продемонстрировано, как веб-сервисы могут изменить способ построения и использования коммерческого приложения. Заинтересованный в закупке коньков оптовый покупатель вводит запрос с помощью собственной локальной системы управления складскими запасами, доступной компьютерам магазинов как веб-сервис. Локальная система управления складскими запасами контактирует через Интернет с веб-сервисом производителя и посылает ему заказ на определенное число коньков, основываясь на объеме имеющихся складских помещений и учитывая наиболее ходовые размеры ботинок.



*Рис. 1.3. Служба ввода заказов Skateboots Company содержит   
несколько других веб-сервисов*

Система ввода заказов компании Skateboots Company состоит из нескольких сервисов, включая специальную часть, которая отвечает уникальным аспектам ее продукции, и несколько торговых компонентов, выполняющих стандартные функции, например проверку полномочий пользователя, проверку кредитоспособности и отслеживание счета. Все эти сервисы предоставляются другими компаниями, специализирующимися на оказании таких услуг через Интернет.

Создание использующих веб-сервисы коммерческих приложений влечет за собой необходимость установления между этими сервисами соответствующих взаимоотношений, что реализуется с помощью любой комбинации языков программирования, операционных систем, пакетов программ, размещенных "до" или "после" брандмауэра. (Это также является способом решения сложных EAI-проблем.) Установление необходимых отношений (или потока управления) связанных веб-сервисов также приводит к автоматизации соответствующих бизнес-процессов и процедур.

Использование веб-сервисов очень выгодно с коммерческой точки зрения. За счет повсеместного распространения веб-сервисов Интернет становится более эффективным, особенно при осуществлении коммерческих сделок. Сочетая прямой доступ к программным приложениям и коммерческим документам, веб-сервисы следующего поколения Сети обеспечат полностью автоматическое взаимодействие, что позволит обращаться непосредственно к данным программ, игнорируя знакомые веб-страницы. Более того, основные компоненты веб-сервисов, скорее всего, будут предоставляться и публиковаться множеством различных компаний, специализирующихся на отдельных функциональных элементах (проверка полномочий, координация сделок, ведение счетов). Это обеспечит непосредственное взаимодействие "приложение-приложение" - принцип, лежащий в основе веб-сервисов и определяющий их суть и реализацию.

**ОБЩЕЕ ВЗАИМОПОНИМАНИЕ**

Технология веб-сервисов существует на очень высоком уровне абстракции, позволяя поддерживать множество одновременных определений, которые иногда бывают противоречивы. На простейшем уровне веб-сервисы могут восприниматься как интернет-ориентированные текстовые брокеры интеграции. Любые данные могут преобразовываться в ASCII-текст и обратно, и этот подход в течение долгого времени был общим знаменателем для систем графического вывода и систем управления базами данных. Как говорится, "если ничего не получается - преобразуй данные в текстовую форму". Ориентированные на использование текста системы также лежат в основе успешного развития Интернета, на котором базируется дополнительная абстракция веб-сервисов. Любой компьютер или операционная система может поддерживать HTML, браузеры и веб-сервисы; и при получении по сети файлов им совершенно безразлично и даже неизвестно, с каким типом прикладной системы они взаимодействуют.

То же самое можно сказать и о веб-сервисах, которые зачастую вызывают замешательство, когда разработчики традиционного, устоявшегося компьютерного окружения пытаются воспринимать веб-сервисы как отдельный тип распределенной программной среды, такой как CORBA, J2EE или .NET. Поскольку веб-сервисы гораздо более абстрактны - как посредники (преобразователи), а не как интерфейсы - так и будет продолжаться, пока они не будут приведены в соответствие с общими описаниями и соглашениями.

**Взаимодействие с веб-сервисами**

Веб-сервисы поддерживают несколько парадигм обмена сообщениями. Уровень абстракции, на котором оперируют веб-сервисы, подразумевает такие стили взаимодействия, как эмуляцию удаленного вызова процедуры (Remote Procedure Call, RPC), асинхронный обмен сообщениями, однонаправленную передачу сообщений, широковещание и публикацию/подписку. Основные СУБД, такие как Oracle, SQL Server и DB2, поддерживают анализ XML и службы преобразования, обеспечивая непосредственное взаимодействие между веб-сервисами и СУБД. Производители связующего программного обеспечения обычно также предоставляют возможность привязки веб-сервисов к своим программным системам (серверам приложений и брокерам интеграции). Следовательно, для пользователя взаимодействие с веб-сервисами может проявляться в интерактивной или пакетной форме, поддерживающей синхронную и асинхронную модели связи; а также как пользовательский интерфейс, написанный с использованием Java, VB (Visual Basic), офисных приложений, браузеров или "толстых" клиентов СУБД. Такое взаимодействие может привязываться к любому типу базовой (более низкого уровня) программной системы.

Веб-сервисы выполняют RPC- и документно-ориентированное взаимодействия. Стандарты и технологии веб-сервисов обычно подразумевают два основных типа моделей взаимодействия приложений:

 удаленный вызов процедуры (онлайновая);

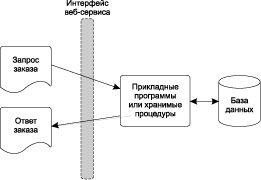
 документно-ориентированный (пакетная).

Эти два типа взаимодействия мы и рассмотрим в последующих разделах.

**RPC-ориентированные взаимодействия**

RPC-ориентированные взаимодействия удобны для краткого обмена данными. В RPC-ориентированном взаимодействии запросы веб-сервисов приобретают форму вызова метода или процедуры с соответствующими входными или выходными параметрами. В отличие от документно-ориентированного взаимодействия, RPC-ориентированное взаимодействие производит отправку документа, специально отформатированного для передачи в отдельную логическую программу или базу данных (рис. 1.4). Поскольку, например, заказ коньков в режиме реального времени зависит от их наличия на складе, программа обращается к базе данных с проверкой наличия заказываемого товара. Если получено подтверждение, то программа вернет заказчику XML-документ в формате "запрос/ответ", сообщающий о принятии заказа с его последующим исполнением.

Если поставка невозможна, будет получено сообщение об отсрочке выполнения заказа или о полном отказе от его выполнения. В отличие от документно-ориентированного стиля взаимодействия, запрос и ответ моделируются как синхронные сообщения, то есть приложение, посылающее сообщение, ждет реакции на него.

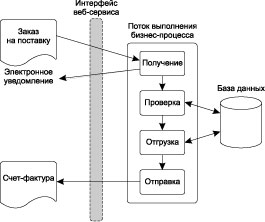


*Рис. 1.4. Веб-сервисы поддерживают интерактивный   
заказ в форме запроса/ответа*

**Документно-ориентированные взаимодействия**

Документно-ориентированные взаимодействия удобны для обмена большими объемами данных. При документно-ориентированном взаимодействии запросы веб-сервиса имеют форму завершенного XML-документа, предназначенного для обработки целиком. Например, веб-сервис, который представляет заказ на поставку (оптовый предварительный заказ на поставку коньков), должен сразу предъявлять производителю полную форму заказа (рис. 1.5). Это напоминает помещение сообщения в очередь для асинхронной обработки. Производитель обычно по электронной почте либо в какой-либо иной форме высылает заказчику подтверждение, свидетельствующее о том, что заказ принят и будет выполнен в соответствии с предопределенной последовательностью выполнения бизнес-процесса. Последовательность выполнения может содержать такие этапы, как проверка в базе данных предыдущих заказов данного оптового покупателя на предмет исчерпания кредитного лимита, согласованных объемов или графика поставок по данному заказу. В реальном потоке выполнения заказа и отправки счета, конечно же, существуют и другие этапы, но в данном примере рассматривается только заключительная стадия: отправка XML-счета для оплаты после доставки и получения заказа.

Необходимые взаимодействия определяются в договоре торговых партнеров. Документно-ориентированные взаимодействия зачастую предполагают, что использующие веб-сервисы стороны заранее согласовали порядок оформления общих документов, таких как заказ на приобретение, счет за доставку или общий счет. Эти стороны обычно идентифицируются как "торговые партнеры" или "сотрудничающие партнеры". Торговые партнеры также обыкновенно согласовывают общий поток выполнения процесса или модель взаимодействия при обмене документами, например, оговаривают необходимость подтверждения квитанции заказа на приобретение, передачу специальной информации о состоянии в ответ на запрос заказа или отправку сигнала оповещения по электронной почте после отгрузки заказа. В ходе реализации бизнес-процесса необходим обмен полными документами. Если до этого документ содержал общую, фрагментированную информацию, то теперь требуется согласованное заполнение специальных разделов, таких как цена покупки или обязательная дата доставки.



*Рис. 1.5. Веб-сервис обрабатывает полный заказ на поставку*

Для синхронных/асинхронных парадигм обмена сообщений удобны два стиля привязки. В нашем примере предварительный оптовый заказ выполняется в компании Skateboots Company с использованием заказов на покупку, представляемых в виде пакетов в соответствии с предварительными договоренностями, что помогает производителю спланировать объемы. В период активного спроса заказы на немедленное пополнение обрабатываются более интерактивными сервисами, которые зависят от заказов на пополнение с имеющихся складов и позволяют тотчас идентифицировать отложенные заказы. Таким образом, сайт Skateboots.com предоставляет веб-сервисы, поддерживающие оба основных типа взаимодействия.

**Технология веб-сервисов**

Порядок описания, поиска и взаимодействия веб-сервисов друг с другом определяют стандарты. Взаимодействующие через Интернет программы должны уметь обнаруживать друг друга, находить информацию, позволяющую им осуществить связь, понимать, какая модель контактирования должна быть применена (простая, типа "запрос/ответ", или более сложная последовательность), и договариваться об использовании таких услуг, как защита информации, подтверждение передачи сообщений и составление сделок. Некоторые из этих сервисов реализуются существующими технологиями и предлагаемыми стандартами, а другие - нет. Использующее веб-сервисы сообщество стремится удовлетворить все эти требования, но это - эволюционный процесс, как и сам Интернет. С самого начала инфраструктура и стандарты веб-сервисов подразумевали возможность расширения (так же как до них XML и HTML), что позволяет использовать их сразу же после появления новых стандартов и технологий.

Веб-сервисы требуют использования нескольких смежных XML-технологий. Для транспортировки и преобразования данных в программы и обратно веб-сервисы требуют использования нескольких смежных XML-технологий.

 Язык XML (Extensible Markup Language) - фундамент, на котором строятся веб-сервисы. Он предоставляет язык определения данных и порядок их обработки. XML представляет семейство связанных спецификаций, публикуемых и поддерживаемых интернет-консорциумом (World Wide Web Consortium, W3C) и другими организациями.

 WSDL (Web Services Description Language) - технология, основанная на XML, определяющая интерфейсы веб-сервисов, типы данных и сообщений, а также модели взаимодействия и протоколы связывания.

**НОВАЯ "СЕРЕБРЯНАЯ ПУЛЯ"?**

С точки зрения современных проблем вычислительной техники веб-сервисы иногда представляются как решение типа "серебряная пуля", исполняя роль, которая ранее отводилась Сети, реляционным базам данных, языкам программирования 4-го поколения и искусственному интеллекту. К сожалению, сами по себе веб-сервисы не в состоянии решить многие проблемы. Они представляют новый уровень (или другой способ выполнения), а не фундаментальное изменение, которое приходит на смену существующей компьютерной инфраструктуре. Этот новый технологический уровень выполняет новую функцию, но, что более важно, обеспечивает механизм интеграции, определенный на более высоком уровне абстракции.

Важность веб-сервисов заключается в том, что они могут служить "связующим мостом", а не заменой существующих технологий. Можно сказать, что новые языки программирования, такие как Visual Basic, C#, C/C++ и Java, пришли на смену старым языкам, таким как COBOL и FORTRAN, хотя множество программ на этих языках все еще используются, как и привязки веб-сервисов для них. В качестве веб-серверов веб-сервисы являются дополнением и не конфликтуют с существующими приложениями, программами и базами данных. Разработка приложений по-прежнему требует знаний Java, VB и C#. Все, что является новым, - это способ преобразования данных для передачи в приложение и обратно на основе стандартных форматов XML-данных и протоколов, что позволяет обеспечить новый уровень интеграции и взаимодействия.

При проектировании и разработке новых программ и баз данных разработчики могут учитывать веб-сервисы, но эти программы и базы данных все равно потребуют использования упаковщиков веб-сервисов. Веб-сервисы сами по себе не являются исполняемыми; они полагаются на исполняемые программы, написанные с помощью языков программирования или сценариев. Веб-сервисы определяют мощный уровень абстракции, который может применяться для межпрограммного взаимодействия на основе существующей сетевой инфраструктуры, но без этой инфраструктуры они ничего не значат.

 Протокол SOAP (Simple Object Access Protocol) - совокупность XML-технологий, определяющих "конверт" для связи веб-сервисов (привязанный к HTTP и другим транспортам), - специализированный формат для передачи XML-документов по Сети, а также соглашения RPC-взаимодействий.

 Технология UDDI (Universal Description, Discovery and Integration) - реестр веб-сервисов и механизм поиска. Он используется для хранения и упорядочения деловой информации, а также для нахождения указателей на интерфейсы веб-сервисов.

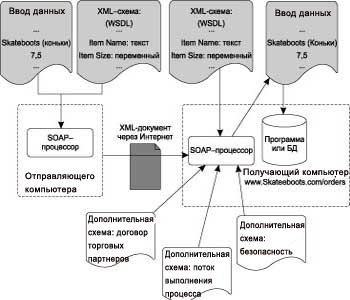
**Пример использования**

Стандарты веб-сервисов обычно используются совместно. Основные стандарты веб-сервисов используются согласованно. После обнаружения WSDL в UDDI или другом месте генерируется SOAP-сообщение для отправки на удаленный сайт.

Как видно из рис. 1.6, при предоставлении документа по адресу веб-сервиса программа использует XML-схему определенного типа (такую как WSDL), позволяющую преобразовать данные из ее входного источника (в этом примере структурированный файл) и на основе того же WSDL-файла создать экземпляр XML-документа в формате, согласованном с целевым веб-сервисом. WSDL-файл используется для определения как входного, так и выходного преобразования данных.

SOAP-процессор отправляющего компьютера преобразует данные из собственного ("родного") формата в тип данных, предопределенный в соответствии с содержащейся в WSDL-файле XML-схемой на основе таблиц преобразования для текстов, значений с плавающей точкой и других данных. Таблицы преобразования "связывают" собственные типы данных с соответствующими конкретной XML-схеме. (Стандартное преобразование типов широко используется в Java, Visual Basic, CORBA и других известных системах. Многие средства XML позволяют настраивать специальные преобразования типов.) SOAP-процессор получающего компьютера выполняет обратное преобразование данных из типов XML-схемы в собственные типы данных.

Файлы описаний веб-сервисов обычно регистрируются с помощью URL. URL, повсеместно используемый в Сети, указывает на IP-адрес, соответствующий веб-ресурсу. Схемы веб-сервисов являются одной из форм веб-ресурса, они содержатся в доступных через Интернет файлах и к ним применим тот же механизм, что используется при загрузке HTML-файлов. Главное отличие между загрузкой HTML-файла и обращением к ресурсу веб-сервиса заключается в том, что веб-сервис оперирует XML-документами, а не HTML-документами и опирается на соответствующие технологии, такие как использование схем, преобразование, проверка подлинности, что и обеспечивает поддержку удаленного соединения приложений. Но способ, согласно которому схемы веб- сервисов публикуются и загружаются, одинаков: HTTP-операция по указанному URL.



*Рис. 1.6. Веб-сервисы используют XML-документы и   
осуществляют преобразование данных*

Для проверки достоверности сообщений веб-сервисы используют XML-схемы. После получения документа реализация веб-сервиса сначала должна проанализировать XML-сообщение и удостовериться в корректности данных, выполнить проверку качества услуг (Quality-of-Service), такую как проверку политики безопасности или соглашений торговых партнеров, а затем произвести последовательность связанных с данным документом коммерческих операций. Веб-сервис на вымышленном нами сайте skateboots.com размещен в папке skateboots.com/ order, на которую и указывает URL.

Веб-сервис, доступный по данному интернет-адресу, идентифицируется с помощью публичного WSDL-файла, который может быть загружен на отправляющий компьютер и использоваться при генерации сообщения. Компания Skateboots Company также осуществляет отправку в общедоступный каталог UDDI-листинга, позволяющего клиентам находить компанию с помощью технологии UDDI. В общем случае, любой, кто хочет взаимодействовать с веб-сервисом, размещающим или контролирующим по Сети заказы для Skateboots Company, для генерации сообщения должен найти способ получения и использования WSDL-файла.

Размещенные по адресу skateboots.com программы представляют собой прослушивающий HTTP-процесс, связанный с соответствующими веб-сервисами для распознавания XML-сообщений, определенных в данном формате. Эти программы включают в себя XML-анализаторы и преобразователи. Кроме того, они осуществляют конвертацию данных SOAP-сообщения в форматы, необходимые для системы ввода заказов компании Skateboots Company.

Технологии веб-сервисов сформировались из основных структур. Этих технологий достаточно для построения, развертывания и публикации базовых веб-сервисов. На самом деле, необходим лишь базовый протокол SOAP. С момента появления веб-сервисов к ним все время добавляются другие технологии. Хотя для деловой связи, а также для создания "моста" с несовместимыми технологиями вполне достаточно базовых принципов, данная форма веб-взаимодействия тем не менее была одобрена очень быстро.

**Основы XML**

Язык XML имеет множество применений. В контексте рассмотрения веб-сервисов язык XML понимается не только как формат сообщений, но и как способ, которым эти сервисы определяются. Следовательно, о самом XML необходимо знать несколько больше, особенно при рассмотрении того, как этот язык используется для определения и реализации веб-сервисов.

**Для чего нужен XML**

XML позволяет определять любое количество элементов. Язык XML был разработан в целях преодоления ограничений языка HTML, в частности для усовершенствования возможностей создания динамического наполнения и управления им. HTML очень удобен для определения и поддержки статического содержания, но не при движении Сети в направлении программной платформы, в которой данные должны генерироваться и систематизироваться динамически. С помощью XML вы можете определить любое множество элементов, значение которых ассоциировано с данными; то есть вы описываете данные и указываете то, что хотите с ними делать, используя один и более элементов, созданных для этой цели. Например:

<Company>

<CompanyNameregion="US">

Skateboots Manufacturing

</CompanyName>

<address>

<line>

200 High Street

</line>

<line>

Springfield, MA 55555

</line>

<Country>

USA

</Country>

</address>

<phone>

+1 781 555 5000

</phone>

</Company>

В этом примере язык XML позволяет описать не только сами элементы данных, но и структуры, которые группируют соответствующие данные. Легко представить поиск элементов, удовлетворяющих точному критерию, например, и нужной компании или по всем элементам с получением списка всех объектов, определяющих себя в Сети как компании.

Кроме того, как уже упоминалось ранее, XML позволяет связывать схемы, что дает возможность раздельно проверять достоверность данных и описывать их прочие атрибуты и качественные характеристики, что невозможно сделать с помощью HTML.

XML-схемы сдерживают гибкость. Разумеется, из-за большой эластичности XML возникают серьезные проблемы. Поскольку язык XML позволяет пользователю определять собственные структурные элементы, сложно гарантировать, что одинаковые элементы будут использоваться одинаковым способом. Для достижения "взаимопонимания" приходится вводить согласованные модели интерпретации содержания.

Две обменивающиеся XML-данными стороны могут однообразно понимать и интерпретировать элементы только в том случае, если они совместно используют одинаковые определения. Если эти же стороны, кроме того, разделяют сообща одну и ту же схему, они будут уверены в однообразном восприятии одинаковых тегов элементов. Именно так и работают веб-сервисы.

**Технологии**

В веб-сервисах используются ряд элементов семейства XML. Язык XML - это семейство технологий: язык разметки данных, различные модели интерпретации содержания, модели связывания, модели пространства имен и различные механизмы преобразования. Ниже представлены наиболее важные элементы семейства XML, являющиеся основой веб-сервисов.

 XML v1.0: правила определения элементов, атрибутов и тегов, включаемых в корневой элемент документа, обеспечивающие абстрактную модель данных и формат сериализации.

 XML schema: XML-документы, определяющие типы данных, содержание, структуру и допустимые элементы в связанных документах. Схемы также используются для описания связанных с элементами документа инструкций семантической обработки.

 XML namespaces (пространство имен XML): однозначно квалифицируемые имена элементов XML-документов и приложений.

 XML Information Set: согласованное абстрактное представление участков XML-документа.

 XPointer (XML Pointer Language), XPath (XML Path Language), XLink (XML Linling Language): средства целеуказания XML-документов.

 Extensible Stylesheet Language Transformations, XSLT: преобразование XML- документов в другие форматы XML-документов или в не-XML форматы.

 DOM (Document Object Model ) и SAX (Simple API for XML): библиотеки и модели программирования для синтаксического разбора XML-документов, который может выполняться либо путем создания всего дерева поиска, либо последовательным чтением и анализом элементов.

**WSDL: описание веб-сервисов**

Язык описания веб-сервисов (Web Services Description Language, WSDL) - это формат XML-схем, определяющий расширенную структуру описания интерфейсов веб-сервисов. WSDL первоначально был разработан компаниями Microsoft и IBM. А затем его поддержали консорциум W3C и 25 компаний. WSDL - это сердце структуры веб-сервиса. Это общий способ представления передаваемых в сообщениях типов данных, указывающий действия, которые должны быть выполнены с данным сообщением и согласно которому сообщения привязываются к сетевым транспортам.

WSDL - это XML-формат, описывающий состав веб-сервиса. WSDL предназначен для использования как в процедурно-ориентированных, так и в документно-ориентированных приложениях. Так же как и другие XML-технологии, WSDL является расширяемым языком и имеет такое количество параметров, что обеспечение совместимости при организации взаимодействия между различными реализациями может вызвать сложности. Полное взаимопонимание возможно лишь в том случае, если отправитель и получатель сообщения могут совместно использовать и одинаково интерпретировать один и тот же WSDL-файл.

WSDL в соответствии с уровнем абстрагирования состоит из трех элементов. WSDL можно разделить на три основные составляющие:

 определение типов данных;

 абстрактные операции;

 связывание сервисов.

Каждая составляющая может быть указана в различных XML-документах и импортирована в виде различных комбинаций, что позволяет создавать окончательное описание веб-сервиса. Кроме того, все составляющие могут быть представлены и в одном XML-документе. "Определение типов данных" задает структуру и содержание сообщений. "Абстрактные операции" определяют операции, которые должны быть выполнены с содержанием сообщения, а "связывание сервисов" подразумевает сетевой транспорт, который доставит сообщение по месту назначения.

На рис. 1.7 представлены составляющие WSDL, упорядоченные в соответствии с их уровнем абстракции, который определяется независимо от транспорта, особенно когда для одного сервиса используется множество транспортов. Например, к одному и тому же сервису можно обращаться посредством "SOAP поверх HTTP" или "SOAP поверх JMS". Аналогично, определения типов данных размещены в разных разделах, что позволяет использовать их различными сервисами. Основные WSDL-составляющие разделены на компоненты.



*Рис. 1.7. WSDL состоит из трех основных составляющих  
и семи компонентов*

WSDL-интерфейсы подобны интерфейсам CORBA и DCOM. Части описания содержат описания типов данных, сообщения и абстрактные операции, которые схожи с интерфейсами CORBA и DCOM. Сообщения могут состоять из нескольких секций и могут быть предназначены для использования в случае процедурно-ориентированного стиля взаимодействия, документно-ориентированного стиля или при применении обоих стилей. В зависимости от уровня абстрагирования одни и те же сообщения могут определяться и использоваться для нескольких типов портов. Как и другие части WSDL, сообщения также поддерживают компоненты расширения, в частности, для включения в себя атрибутов других сообщений.

Типы данных веб-сервисов основаны на XML-схемах, но могут быть распространены на любые другие механизмы. WSDL-описания типов данных базируются на XML-схемах, но здесь годятся для использования и другие эквивалентные или аналогичные системы определения типов данных. Например, вместо типов данных XML-схемы может применяться язык описания интерфейса (Interface Definition Language, IDL) CORBA. (Если используется иная модель определения типов, ее должны "понимать" обе стороны.)

Абстрактные операции и сообщения отображаются на специальный транспорт. Связывание сервисов отображает абстрактные сообщения и операции на специальный транспорт, такой как SOAP. Для включения сведений, характерных для SOAP и других типов отображения, используется связывание расширяемых компонентов. Абстрактные определения могут привязываться к различным физическим транспортам. Спецификация WSDL v1.1 включает примеры однонаправленного SOAP-отображения для SMTP (Simple Mail Transfer Protocol), SOAP RPC-привязки к HTTP, SOAP-привязки к методам get и post протокола HTTP, а также пример отображения для MIME (Multipurpose Internet Mail Extensions) - многокомпонентного связывания SOAP.

Пространство имен гарантирует уникальность имен WSDL-элементов. Пространство имен XML предназначено для обеспечения уникальности имен, используемых в любой из трех основных WSDL-составляющих. Конечно же, при раздельной разработке WSDL-элементов и импортировании их в один файл описанные в отдельных файлах пространства имен не должны перекрываться. Связанные схемы используются для проверки достоверности как WSDL-файла и сообщений, так и для операций, определенных в WSDL-файле.

Можно смело сказать, что по мере развития веб-сервисов WSDL, по всей видимости, будет включать множество расширений, изменений и добавлений. Как и SOAP, WSDL разработан в виде расширяемой XML-структуры, которая может быть легко модифицирована для привязки множества типов данных, определений типов сообщений, операций и транспортов. Например, рабочие группы IETF (Internet Engineering Task Force) предлагают новый стандарт протоколов - Blocks Extensible Exchange Protocol (BEEP), - позволяющий определять транспорт с установлением соединений. (В отличие от HTTP, ориентированного на связь без установления соединения, что усложняет решение проблем качества обслуживания на транспортном уровне.) Компании, интересующиеся использованием веб-сервисов во внутренних приложениях или для осуществления интеграции, могут расширить WSDL, обеспечив его привязку к более традиционным протоколам, таким как DCOM или IIOP (Internet Inter-Orb Protocol).

**SOAP: доступ к веб-сервисам**

До сего момента вы определяли данные (XML) и абстрактно представили службу, необходимую для обеспечения соединения и обработки сообщения (WSDL). Теперь же необходимо уточнить, каким способом сообщение будет пересылаться с одного узла на другой, что позволит провести его обработку на конечном компьютере.

SOAP обеспечивает механизм подключения к веб-сервисам. Спецификация протокола SOAP определяет структуру сообщения, используемого для обмена форматированными XML-данными посредством сети Интернет. Структура сообщения достаточно проста, удобна при разработке и полностью нейтральна по отношению к операционной системе, языку программирования и компьютерной платформе. Протокол SOAP предназначен для обеспечения транспорта базового уровня, на основе которого могут возводиться более сложные протоколы и способы взаимодействия.

SOAP - это XML-способ определения: какая информация должна пересылаться и как. SOAP - это базовая однонаправленная модель соединения, обеспечивающая согласованную передачу сообщения от отправителя к получателю, потенциально допускающая наличие посредников, которые могут обрабатывать часть сообщения или добавлять к нему дополнительные элементы. Спецификация SOAP содержит соглашения по преобразованию однонаправленного обмена сообщениями в соответствии с принципом "запрос/ответ", популярным при организации соединения типа RPC, а также определяет, как осуществлять передачу всего XML-документа.

SOAP включает дополнительные правила кодирования для различных типов данных, но конечные узлы SOAP-соединения могут определить собственные правила посредством конфиденциального соглашения сторон. В ходе связи зачастую используется буквенное или встроенное в XML кодирование.

Как видно из рис. 1.8, SOAP предназначена для поддержания независимого абстрактного протокола связи, обеспечивающего коммуникацию двух и более предприятий или двух и более удаленных корпоративных сайтов. Связанные системы можно построить на основе любой комбинации аппаратного и программного обеспечения, поддерживающей доступ через Интернет к таким средам, как .NET и J2EE. Существующие системы обычно также представлены множественными инфраструктурами и пакетами программ. SOAP и остальные структуры XML предоставляют возможность двум и более предприятиям, торговым площадкам или партнерам выработать общий подход по вопросу предоставления услуг в Сети.

SOAP-сообщения содержат конверт, заголовок и тело сообщения. SOAP-сообщения состоят из нескольких основных частей.

Envelope (конверт) - определяет начало и конец сообщения.

Header (заголовок) - содержит любые дополнительные атрибуты сообщения, используемые в ходе обработки сообщения как посредником, так и конечным получателем.

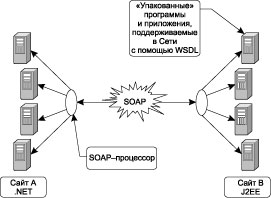
Body (тело сообщения) - содержит XML-данные, передаваемые данным сообщением.

Attachment (вложение) - состоит из одного и более документов, "прикрепленных" к основному сообщению. (Относится только к SOAP with Attachments ("SOAP с вложениями").)

RPC interaction (SOAP:RPC-взаимодействие) - определяет, как моделировать взаимодействия RPC-типа.

Encoding (кодировка) - определяет, как будут представлены простые и сложные данные, передаваемые в сообщении.

Обязательными являются только конверт и тело сообщения.



*Рис. 1.8. SOAP-сообщения осуществляют соединение удаленных узлов*

**UDDI: публикация и поиск веб-сервисов**

После того как вы определили данные в сообщении (XML), описали сервисы, которые будут принимать и обрабатывать сообщения (WSDL), и предъявили требования к средству их передачи и приема (SOAP), вам необходимы способы публикации предлагаемой вами службы, а также инструменты поиска нужных вам, но предоставляемых другими участниками услуг. Именно эти функции и выполняет реестр UDDI (universal distribution, discovery, and integration).

UDDI регистрирует и публикует определения веб-сервисов. Структура UDDI определяет модель данных в программных интерфейсах (API) XML и SOAP для регистрации и обнаружения коммерческой информации, включая веб-сервисы. Реестр UDDI был создан независимым консорциумом производителей (основанным компаниями Microsoft, IBM и Ariba), занимающимся разработкой стандарта для описания, регистрации и локализации веб-сервисов. Microsoft, IBM, Hewlett- Packard и SAP на начальном этапе предоставляют общедоступный UDDI-сервис, который концептуально копирует DNS (систему имен доменов Интернета, осуществляющую "перевод" имен Интернета в IP-адреса). На практике UDDI более похож на сервис реплицируемой базы данных, доступной через Интернет.

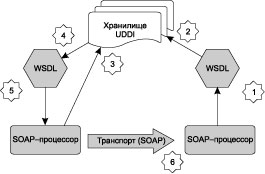
UDDI - это каталог веб-сервисов. UDDI идейно подобен каталогу "Желтые страницы". Предприниматели регистрируют свою контактную информацию, в том числе номер телефона, факса, почтовый адрес и адрес веб-сайта. Регистрация включает различные необходимые для поиска категории информации, такие как географическое положение, код отрасли промышленности, тип бизнеса и т. д. Другие предприниматели при поиске поставщиков комплектующих, служб доставки, а также аукционов и торгов могут найти нужные данные, сохраненные в UDDI. Можно также обнаружить информацию об определенном веб-сервисе обычным путем, пытаясь найти URL WSDL-файла, указывающего на веб-сервис поставщика.

Для регистрации и поиска информации UDDI использует SOAP. Для регистрации предпринимателей в UDDI служит SOAP; далее клиенты реестра для поиска информации необходимого им торгового партнера используют программный интерфейс SOAP-запроса. Первоначальный запрос может вернуть несколько соответствий, из которых может быть выбрано только одно. После выбора необходимого элемента для получения необходимой для осуществления сделки конкретной контактной информации производится заключительный вызов API.

На рис. 1.9 показано, как предприятие будет осуществлять регистрацию в UDDI сведений о веб-сервисе.

Сначала предприятие генерирует WSDL-файл, который описывает веб-сервисы, поддерживаемые SOAP-процессором (1), и с помощью API UDDI регистрирует информацию в хранилище (2). После отправки предприятием этих сведений (совместно с другой контактной информацией) элемент реестра будет содержать URL, указывающий на WSDL-файл сайта SOAP-сервера или на другой файл XML-схемы, описывающий данный веб-сервис.

После этого SOAP-процессор другого предприятия запрашивает реестр (3) для получения WSDL или другой схемы (4). Для отправки конкретной операции по определенному протоколу (6) клиент может сгенерировать соответствующее сообщение (5). Конечно же, клиент и сервер должны договориться о том, чтобы протокол был одинаковым (в данном примере "SOAP поверх HTTP"), и совместно использовать одинаковые понятия или семантические описания сервиса, который в данном примере представлен WSDL. При повсеместном распространении основных стандартов, скорее всего, общее понимание будет гарантироваться посредством WSDL.



*Рис. 1.9. Для обнаружения веб-сервиса может   
использоваться репозиторий UDDI*