

**Міністерство освіти і науки України**  
**ЗНТУ**

Кафедра програмних засобів

## **Архітектури та технології WEB-служб**

**Конспект лекцій**  
для студентів спеціальності

121            Інженерія програмного забезпечення  
усіх форм навчання

# Веб-сервисы как средство интеграции приложений в WWW

## Что есть веб-сервис?

Всемирная паутина является готовой платформой для создания и использования распределенных машинно-ориентированных систем на основе веб-сервисов. Веб-сервер выступает в качестве сервера приложений, к которым обращаются не конечные пользователи, а сторонние приложения. Это позволяет многократно использовать функциональные элементы, устранить дублирование кода, упростить решение задач интеграции приложений.

Веб-служба, **веб-сервис** (англ. web-service) — это сетевая технология, обеспечивающая межпрограммное взаимодействие на основе веб-стандартов. Консорциум [W3C](#) определяет веб-сервис, как «программную систему, разработанную для поддержки интероперабельного межкомпьютерного (machine-to-machine) взаимодействия через сеть»

## Веб-службы: концепции и протоколы

Веб-сервис идентифицируется строкой URI. Веб-сервис имеет программный интерфейс, представленный в машинно-обрабатываемом формате [WSDL](#). Другие системы взаимодействуют с этим веб-сервисом путем обмена сообщениями протокола [SOAP](#). В качестве транспорта для сообщений используется протокол [HTTP](#). Описание веб-сервисов и их API могут быть найдены средствами [UDDI](#). Концептуальная схема технологии приведена на [рис. 1](#), а связь между протоколами — на [рис. 2](#).



Рис. 1. Концепция веб-сервиса

- **SOAP** (Simple Object Access Protocol) — протокол обмена сообщениями между потребителем и поставщиком веб-сервиса;
- **WSDL** (Web Services Description Language) — язык описания внешних интерфейсов веб-службы;
- **UDDI** (Universal Discovery, Description and Integration) — универсальный интерфейс распознавания, описания и интеграции, используемый для формирования каталога веб-сервисов и доступа к нему.



## Рис. 2. Протоколы веб-сервисов

Все спецификации, используемые в технологии, основаны на [XML](#) и, соответственно, наследуют его преимущества (структурированность, гибкость и т.д.) и недостатки (громоздкость, медлительность).

### SOAP

SOAP (изначально *Simple Object Access Protocol*, а в версии 1.2 официальная расшифровка аббревиатуры отсутствует) — простой протокол доступа к объектам (компонентам распределенной вычислительной системы), основанный на обмене структурированными сообщениями. Как любой текстовый протокол, SOAP может использоваться с любым протоколом прикладного уровня: SMTP, FTP, HTTPS и др., но чаще всего SOAP используется поверх HTTP.

Все сообщения SOAP оформляются в виде структуры, называемой *конвертом* (envelop), включающей следующие элементы:

- Идентификатор сообщения (локальное имя).
- Опциональный элемент Header (заголовок):
  - Ноль или более ссылок на используемые пространства имен;
  - Ноль или более свойств, доступных в этом пространстве имен.
- Обязательный элемент Body (тело сообщения)
  - Ноль или более ссылок на используемые пространства имен;
  - Дочерние элементы тела сообщения

Развернутый список элементов сообщения SOAP приведен в [схеме данных](#) (для SOAP версии 1.2).

Пример сообщения SOAP:

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
  <env:Header>
    <n:alertcontrol xmlns:n="http://example.org/alertcontrol">
      <n:priority>1</n:priority>
      <n:expires>2001-06-22T14:00:00-05:00</n:expires>
    </n:alertcontrol>
  </env:Header>
  <env:Body>
    <m:alert xmlns:m="http://example.org/alert">
      <m:msg>Get up at 6:30 AM</m:msg>
    </m:alert>
  </env:Body>
</env:Envelope>
```

### XML-RPC: Не конкурент, а альтернатива SOAP

XML-RPC — очень простой и эффективный протокол взаимодействия веб-сервисов. Он не предназначен для решения глобальных задач, как SOAP, но широко используется во многих веб-разработках.



Концепция XML-RPC

XML-RPC — это "... [спецификация](#) и набор реализаций, которые позволяют программному обеспечению, работающему на разных операционных системах и в различных условиях,

вызывать процедуры через Интернет. Это удаленный вызов процедуры с использованием HTTP как транспорта и XML как способа кодирования. XML-RPC разработан настолько простым, насколько это возможно для сложных структур данных, подлежащих передаче, обработке и приему". — [xmlrpc.com]

"Мы хотели, чистый, расширяемый и очень простой формат. Он должен представлять HTML-кодеру возможность заглянуть в файл, содержащий описание XML-RPC вызова, понять, что тот делает и быть в состоянии изменить его, чтоб он заработал с первой или второй попытки... Мы также хотели, чтобы это был легко реализуемый протокол, который может быть быстро адаптирован для работы в другой среде или на других операционных системах."- [xmlrpc.com]

## WSDL

Язык описания веб-сервисов (*Web services Description Language*, WSDL) предназначен для унифицированного представления внешних интерфейсов веб-службы. Текущая версия протокола (на момент написания этой лекции) [WSDL 2.0](#) и она имеет некоторые отличия от предыдущих версий (см. [табл. 1](#) и [рис. 3](#)).

Таблица 1. Основные элементы протокола WSDL.

Элемент WSDL 1.1	Элемент WSDL 2.0	Краткое описание
PortType	Interface	Представляет описание интерфейса веб-сервиса (список операций и их параметров).
Service	Service	Список системных функций
Binding	Binding	Специфицирует интерфейсы и задает параметры связывания с протоколом SOAP: стиль связывания (RPC/Document) и транспорт (SOAP). Эта секция доступна и для каждой из операций
Operation	Operation	Определяет операцию, представляемую веб-сервером. WSDL-операция — это аналог традиционным функциям и процедурам.
Message	не использ.	Сообщение, связанное с определенной операцией. Содержит информацию, необходимую для выполнения данной операции. Каждое сообщение может состоять из нескольких логических частей, описывающих типы данных и имена атрибутов. В версии 2.0 было исключено, т.к. была внедрена поддержка XML Schema для всех элементов.
Types	Types	Описание данных в соответствии с XML Schema.

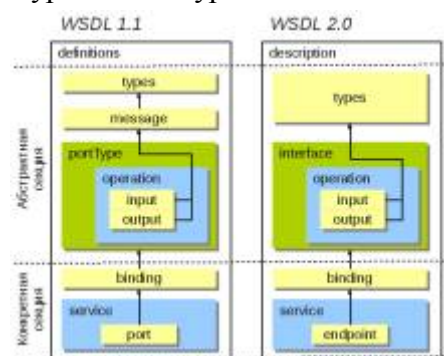


Рис. 3. Структура протокола WSDL

В спецификации WSDL 1.1 было определено 4 шаблона обмена сообщениями (типы операций):

- Однонаправленные операции (One-way): операция может принимать сообщение, но не будет возвращать ответ.
- Запрос-ответ (Request-response): операция может принять запрос и должна вернуть ответ.
- Вопрос-ответ (Solicit-response): операция может послать запрос и будет ждать ответ на него.
- Извещение (Notification): операция может послать сообщение, но не будет ожидать ответ.

В версии WSDL 2.0 эти шаблоны изменены и расширены в сторону поддержки сообщений об ошибках (например, шаблон Robust-in-only), но для обратной совместимости поддерживаются типы WSDL 1.1

[Пример описания веб-сервиса на языке WSDL](#) (версия 2.0).

## UDDI

*Universal Description, Discovery and Integration* (UDDI, универсальный интерфейс распознавания, описания и интеграции) — открытый стандарт, утвержденный OASIS, определяющий методы публикации и обнаружения сетевых программных компонентов сервис-ориентированной архитектуры ([SOA](#)). В практической реализации UDDI представляет собой сетевой реестр ([службу каталогов](#)), представляющий данные и метаданные о веб-сервисах и доступный по адресу <http://uddi.xml.org/services>.

UDDI опирается на отраслевые стандарты HTTP, XML, XML Schema (XSD), SOAP и WSDL. Концептуальная связь между UDDI и другими протоколами стека веб-сервисов показана на [рисунке 4](#).



Рис. 4. Место UDDI в стеке протоколов веб-служб

Функциональное назначение реестра UDDI — представление данных и метаданных о веб-службах. Он может использоваться как в сети общего пользования, так и в пределах внутренней инфраструктуры организации. Реестр UDDI предлагает основанный на стандартах механизм классификации, каталогизации и управления веб-службами, позволяющий применять их (веб-сервисы) другими приложениями. Этот механизм включает средства для поиска сервиса, вызова этой службы, а также для управления метаданными об этой службе.

Ключевыми функциями UDDI являются публикация информации о службе в реестре и поиск этой информации сторонними приложениями. Наряду с этими, реализованы и типовые для службы каталогов функции: представление модели хранимых данных и структуры

информационной базы, отношения между объектами реестра, репликация, обеспечение безопасности и т.д. — Все основные функции реестра представлены в виде веб-сервисов и доступны через API UDDI.

## Сервис www

Интернет — это глобальная компьютерная сеть, объединяющая сотни миллионов компьютеров в общее информационное пространство. Интернет представляет свою инфраструктуру для прикладных сервисов различного назначения, самым популярным из которых является Всемирная Паутина – World Wide Web (www).

World Wide Web (www, web, рус.: веб, Всемирная Паутина) — распределенная информационная система, предоставляющая доступ к гипертекстовым документам по протоколу HTTP.

WWW — сетевая технология прикладного уровня [стека TCP/IP](#), построенная на [клиент-серверной архитектуре](#) и использующая инфраструктуру Интернет для взаимодействия между сервером и клиентом ([рис. 1](#)).

Серверы www (*веб-серверы*) — это хранилища гипертекстовой (в общем случае) информации, управляемые специальным программным обеспечением.

Документы, представленные в виде гипертекста называются *веб-страницами*. Несколько веб-страниц, объединенных общей тематикой, оформлением, связанных гипертекстовыми ссылками и обычно находящихся на одном и том же веб-сервере, называются *веб-сайтом*.

Для загрузки и просмотра информации с веб-сайтов используются специальные программы — *браузеры*, способные обрабатывать гипертекстовую разметку и отображать содержимое веб-страниц.

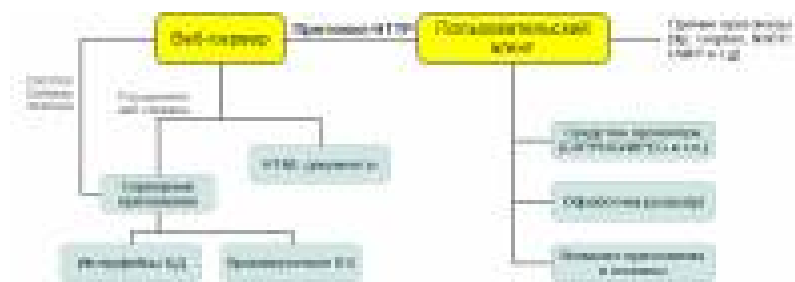


Рис. 1. Архитектура сервиса WWW

В основе www — взаимодействие между веб-сервером и браузерами по [протоколу HTTP](#) (HyperText Transfer Protocol). Веб-сервер — это программа, запущенная на сетевом компьютере и ожидающая клиентские запросы по протоколу HTTP. Браузер может обратиться к веб-серверу по доменному имени или по ip-адресу, передавая в запросе идентификатор требуемого ресурса. Получив запрос от клиента, сервер находит соответствующий ресурс на локальном устройстве хранения и отправляет его как ответ. Браузер принимает ответ и обрабатывает его соответствующим образом, в зависимости от типа ресурса (отображает гипертекст, показывает изображения, сохраняет полученные файлы и т.п.).

Основной тип ресурсов Всемирной паутины — гипертекстовые страницы. Гипертекст — это обычный текст, размеченный специальными управляющими конструкциями — тегами. Браузер считывает теги и интерпретирует их как команды форматирования при выводе

информации. Теги описывают структуру документа, а специальные теги, якоря и гиперссылки, позволяют установить связи между веб-страницами и перемещаться как внутри веб-сайта, так и между сайтами.

Сэр [Тимоти Джон Бернерс-Ли](#) — британский учёный-физик, изобретатель Всемирной паутины (совместно с Робертом Кайо), автор URI, HTTP и HTML. Действующий глава Консорциума Всемирной паутины (W3C). Автор концепции семантической паутины и множества других разработок в области информационных технологий. 16 июля 2004 года Королева Великобритании Елизавета II произвела Тима Бернерса-Ли в Рыцари-Командоры за «службу во благо глобального развития Интернета».

## Компоненты сервиса

Функционирование сервиса обеспечивается четырьмя составляющими:

- [URL/URI](#) — унифицированный способ адресации и идентификации сетевых ресурсов;
- [HTML](#) — язык гипертекстовой разметки веб-документов;
- [HTTP](#) — протокол передачи гипертекста;
- [CGI](#) — общий шлюзовый интерфейс, представляющий доступ к серверным приложениям.

## Адресация веб-ресурсов. URL, URN, URI

Для доступа к любым сетевым ресурсам необходимо знать где они размещены и как к ним можно обратиться. Во Всемирной паутине для обращения к веб-документам изначально используется стандартизованная схема адресации и идентификации, учитывающую опыт адресации и идентификации таких сетевых сервисов, как e-mail, telnet, ftp и т.п. — URL, Uniform Resource Locator.

URL ([RFC 1738](#)) — унифицированный локатор (указатель) ресурсов, стандартизированный способ записи адреса ресурса в www и сети Интернет. Адрес URL имеет гибкую и расширяемую структуру для максимально естественного указания местонахождения ресурсов в сети. Для записи адреса используется ограниченный набор символов ASCII. Общий вид адреса можно представить так:

```
<схема>://<логин>:<пароль>@<хост>:<порт>/<полный-путь-к-ресурсу>
```

Где:

схема

схема обращения к ресурсу: http, ftp, gopher, mailto, news, telnet, file, man, info, whatis, ldap, wais и т.п.

логин:пароль

имя пользователя и его пароль, используемые для доступа к ресурсу

хост

[доменное имя](#) хоста или его IP-адрес.

порт

порт хоста для подключения

полный-путь-к-ресурсу

уточняющая информация о месте нахождения ресурса (зависит от протокола).

Примеры URL:

`http://example.com`      #запрос стартовой страницы по умолчанию

```
http://www.example.com/site/map.html #запрос страницы в указанном каталоге
http://example.com:81/script.php #подключение на нестандартный порт
http://example.org/script.php?key=value #передача параметров скрипту
ftp://user:pass@ftp.example.org #авторизация на ftp-сервере
http://192.168.0.1/example/www #подключение по ip-адресу
file:///srv/www/htdocs/index.html #открытие локального файла
gopher://example.com/1 #подключение к серверу gopher
mailto://user@example.org #ссылка на адрес эл.почты
```

В августе 2002 года [RFC 3305](#) анонсировал устаревание URL в пользу URI (Uniform Resource Identifier), еще более гибкого способа адресации, вобравшего возможности как URL, так и URN (Uniform Resource Name, унифицированное имя ресурса). URI позволяет не только указывать местонахождение ресурса (как URL), но и идентифицировать его в заданном пространстве имен (как URN). Если в URI не указывать местонахождение, то с его помощью можно описывать ресурсы, которые не могут быть получены непосредственно из Интернета (автомобили, персоны и т.п.). Текущая структура и синтаксис URI регулируется стандартом [RFC 3986](#), вышедшим в январе 2005 года.

## Язык гипертекстовой разметки HTML

HTML ([HyperText Markup Language](#)) — стандартный язык разметки документов во Всемирной паутине. Большинство веб-страниц созданы при помощи языка HTML. Язык HTML интерпретируется браузером и отображается в виде документа, в удобной для человека форме. HTML является приложением SGML (стандартного обобщённого языка разметки) и соответствует международному стандарту ISO 8879.

HTML создавался как язык для обмена научной и технической документацией, пригодный для использования людьми, не являющимися специалистами в области вёрстки. Для этого он представляет небольшой (сравнительно) набор структурных и семантических элементов — тегов. С помощью HTML можно легко создать относительно простой, но красиво оформленный документ. Изначально язык HTML был задуман и создан как средство структурирования и форматирования документов без их привязки к средствам воспроизведения (отображения). В идеале, текст с разметкой HTML должен единообразно воспроизводиться на различном оборудовании (монитор ПК, экран органайзера, ограниченный по размерам экран мобильного телефона, медиа-проектор). Однако современное применение HTML очень далеко от его изначальной задачи. Со временем основная идея платформонезависимости языка HTML стала жертвой коммерциализации www и потребностей в мультимедийном и графическом оформлении.

## Протокол HTTP

HTTP ([HyperText Transfer Protocol](#)) — протокол передачи гипертекста, текущая версия HTTP/1.1 ([RFC 2616](#)). Этот протокол изначально был предназначен для обмена гипертекстовыми документами, сейчас его возможности существенно расширены в сторону передачи двоичной информации.

HTTP — типичный клиент-серверный протокол, обмен сообщениями идёт по схеме «запрос-ответ» в виде ASCII-команд. Особенностью протокола HTTP является возможность указать в запросе и ответе способ представления одного и того же ресурса по различным параметрам: формату, кодировке, языку и т. д. Именно благодаря возможности указания способа кодирования сообщения клиент и сервер могут обмениваться двоичными данными, хотя данный протокол является символьно-ориентированным.



HTTP — протокол прикладного уровня, но используется также в качестве «транспорта» для других прикладных протоколов, в первую очередь, основанных на [языке XML](#) (SOAP, XML-RPC, SiteMap, RSS и проч.).

## Общий шлюзовый интерфейс CGI

CGI ([Common Gateway Interface](#)) — механизм доступа к программам на стороне веб-сервера. Спецификация CGI была разработана для расширения возможностей сервиса www за счет подключения различного внешнего программного обеспечения. При использовании CGI веб-сервер представляет браузеру доступ к исполнимым программам, запускаемым на его (серверной) стороне через стандартные потоки ввода и вывода.

Интерфейс CGI применяется для создания динамических веб-сайтов, например, когда веб-страницы формируются из результатов запроса к базе данных. Сейчас популярность CGI снизилась, т.к. появились более совершенные альтернативные решения (например, модульные расширения веб-серверов).

## Программное обеспечение сервиса www

### Веб-серверы

**Веб-сервер** — это сетевое приложение, обслуживающее HTTP-запросы от клиентов, обычно веб-браузеров. Веб-сервер принимает запросы и возвращает ответы, обычно вместе с HTML-страницей, изображением, файлом, медиа-поток или другими данными. Веб-серверы — основа Всемирной паутины. С расширением спектра сетевых сервисов веб-серверы все чаще используются в качестве шлюзов для [серверов приложений](#) или сами представляют такие функции (например, [Apache Tomcat](#)).

Созданием программного обеспечения веб-серверов занимаются многие разработчики, но наибольшую популярность (по статистике <http://netcraft.com>) имеют такие программные продукты, как Apache ([Apache Software Foundation](#)), IIS ([Microsoft](#)), Google Web Server (GWS, Google Inc.) и [nginx](#).

**Apache** — [свободное программное обеспечение](#), распространяется под совместимой с GPL лицензией. Apache уже многие годы является лидером по распространенности во Всемирной паутине в силу своей надежности, гибкости, масштабируемости и безопасности.

**IIS** (Internet Information Services) — проприетарный набор серверов для нескольких служб Интернета, разработанный Майкрософт и распространяемый с серверными операционными системами семейства Windows. Основным компонентом IIS является веб-сервер, также поддерживаются протоколы FTP, POP3, SMTP, NNTP.

**Google Web Server (GWS)** — разработка компании Google на основе веб-сервера Apache. GWS оптимизирован для выполнения приложений сервиса Google Applications.

**nginx [engine x]** — это HTTP-сервер, совмещенный с кэширующим прокси-сервером. Разработан [И. Сыроевым](#) для компании Рамблер. Осенью 2004 года вышел первый публично доступный релиз, сейчас nginx используется на 9-12% веб-серверов.

### Браузеры

**Браузер, веб-обозреватель (web-browser)** — клиентское приложение для доступа к веб-серверам по протоколу HTTP и просмотра веб-страниц. Как правило браузеры дополнительно поддерживают и ряд других протоколов (например ftp, file, mms, pop3).

Первые HTTP-клиенты были консольными и работали в текстовом режиме, позволяя читать гипертекст и перемещаться по ссылкам. Сейчас консольные браузеры (такие, как lynx, w3m или links) практически не используются рядовыми посетителями веб-сайтов. Тем не менее такие браузеры весьма полезны для веб-разработчиков, так как позволяют «увидеть» веб-страницу «глазами» поискового робота.

Исторически первым браузером в современном понимании (т.е. с графическим интерфейсом и т.д.) была программа NCSA Mosaic, разработанная Марком Андерисеном и Эриком Бина. Mosaic имел довольно ограниченные возможности, но его открытый исходный код стал основой для многих последующих разработок.

Существует большое число программ-браузеров, но наибольшей популярностью пользуются следующие (рис. 2):

Рис. 2. Популярные браузеры (по данным [statcounter.com](http://statcounter.com))

**Internet Explorer (IE)** — браузер, разработанный компанией Майкрософт и тесно интегрированный с ОС Windows. Платформозависим (поддержка сторонних ОС прекращена, начиная с версии 5). Единственный браузер, напрямую поддерживающий технологию ActiveX. Не полностью совместим со стандартами W3C, в связи с чем требует дополнительных затрат от веб-разработчиков.

**Firefox** — свободный кроссплатформенный браузер, разрабатываемый Mozilla Foundation и распространяемый под тройной лицензией GPL/LGPL/MPL. В основе браузера — движок Gecko, который изначально создавался для Netscape Communicator. Однако, вместо того, чтобы предоставить все возможности движка в стандартной поставке, Firefox реализует лишь основную его функциональность, предоставляя пользователям возможность модифицировать браузер в соответствии с их требованиями через поддержку расширений (add-ons), тем оформления и плагинов.

**Safari** — проприетарный браузер, разработанный корпорацией Apple и входящий в состав операционной системы Mac OS X. Бесплатно распространяется для операционных систем семейства Microsoft Windows. В браузере используется уникальный по производительности интерпретатор JavaScript и еще ряд интересных для пользователя решений, которые отсутствуют или не развиты в других браузерах.

**Chrome** — кроссплатформенный браузер с открытым исходным кодом, разрабатываемый компанией Google. Первая стабильная версия вышла 11 декабря 2008 года. В отличие от многих других браузеров, в Chrome каждая вкладка является отдельным процессом. В случае если процесс обработки содержимого вкладки зависнет, его можно будет завершить без риска потери данных других вкладок. Еще одна особенность — интеллектуальная адресная строка (Omnibox). К возможности автозаполнения она добавляет поисковые функции с учетом популярности сайта, релевантности и пользовательских предпочтений (истории переходов).

**Opera** — кроссплатформенный многофункциональный веб-браузер, впервые представленный в 1994 году группой исследователей из норвежской компании Telenor. Дальнейшая разработка ведется Opera Software ASA. Этот браузер обладает высокой скоростью работы и совместим с основными стандартами. Отличительными особенностями Opera долгое время являлись многостраничный интерфейс и возможность масштабирования

веб-страниц целиком. На разных этапах развития в Opera были интегрированы возможности почтового/новостного клиента, адресной книги, клиента сети BitTorrent, агрегатора RSS, клиента IRC, менеджера зачек, WAP-браузера, а также поддержка виджетов — графические модулей, работающих вне окна браузера.

## Роботы-«пауки»

Наряду с браузерами, ориентированными на пользователя, существуют и специализированные клиенты-роботы («пауки», «боты»), подключающиеся к веб-серверам и выполняющие различные задачи автоматической обработки гипертекстовой информации. Сюда относятся, в первую очередь, роботы поисковых систем, таких как google.com, yandex.ru, yahoo.com и т.п., выполняющие обход веб-сайтов для последующего построения поискового индекса.

По мнению ведущих IT-компаний и аналитиков (наиболее известными из них являются IBM, Microsoft, Sun Microsystems, BEA, SAP, Oracle, Gartner Group, , Stencil Group, International Data Corp.), важными и перспективными направлениями в развитии информационных систем и ПО являются архитектуры, ориентированные на сервисы (Service Oriented Architecture - **SOA**) [1, 2]. При этом основной акцент делается на SOA, которая ориентирована на Internet и intranet, т.е. на архитектуру веб-сервисов (Web Services Architecture - **WSA**).

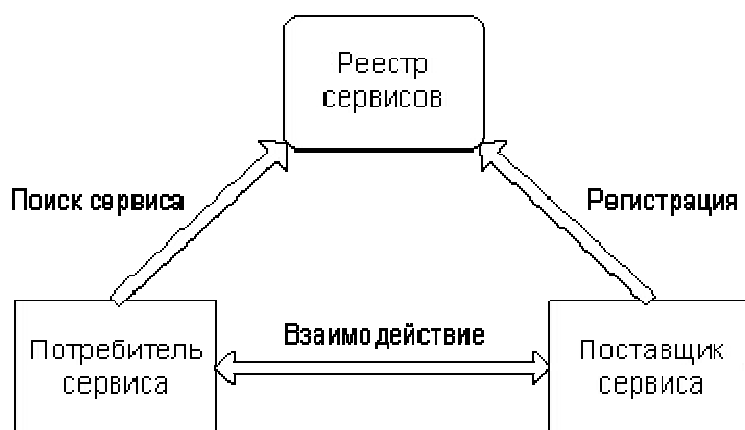
Архитектура, ориентированная на сервисы (**SOA**), имеет следующие характерные особенности:

1. Архитектура является распределенной. Функциональные модули приложение (системы) могут быть распределены по множеству вычислительных систем и способны к взаимодействию с использованием локальных или глобальных сетей.
2. Интерфейс функциональных модулей таков, что использование модулей не зависит от технологии или платформы, в рамках которой они реализованы.
3. Возможен динамический поиск и подключение нужных функциональных модулей.
4. Архитектура базируется на общепринятых отраслевых стандартах.

Основу архитектуры, ориентированной на сервисы, составляет взаимодействие трех участников:

- поставщика сервиса;
- потребителя сервиса;
- реестра сервисов.

Это взаимодействие представлено на рисунке 1.



## Рисунок 1. Компоненты архитектуры SOA

Взаимоотношение участников включает следующие основные аспекты:

- публикация сервиса;
- поиск сервиса;
- подключение и использование.

Для реализации SOA необходимы три типа соглашения:

1. Транспортное соглашение: о форматах и протоколах взаимодействия.
2. Соглашение об описании функциональности сервиса, в виде, пригодном для автоматической генерации кода, который определяет процесс взаимодействия между клиентом и поставщиком сервиса.
3. Соглашение о способе обнаружения сервиса.

Архитектура веб-сервисов является одной из реализаций SOA. Понятие архитектуры, ориентированной на сервисы, сложилось в ходе развития концепции веб-сервисов. Однако, существуют и другие походы к реализации SOA: Java RMI (от Sun Microsystems), CORBA (от консорциума OMG), DCOM (от Microsoft), DCE (предложенный ассоциацией Open Group).

*Концепция веб-сервисов* возникла в конце 90-х годов XX века. Однако, к настоящему моменту эта концепция успела устояться и архитектура, которую она предлагает, стала отраслевым стандартом в сфере ИТ.

Стандартизацией архитектуры веб-сервисов занимаются рабочие группы комитета W3C [5]. Они предлагают следующее определение понятия "*веб-сервис*": "веб-сервис - это реализуемая программными средствами система для поддержки межмашинного взаимодействия через сеть. Интерфейс сервиса описывается на языке, читаемом машиной, например, **WSDL**. Другие системы взаимодействуют с веб-сервисом способом, указанным в его описании, используя сообщения в стандарте **SOAP**, передаваемые с использованием **HTTP** и **XML** и в сочетании с другими стандартами, относящимися к Web".

Физически Web-сервис представляет собой фрагмент программного обеспечения, называемый "*агентом*". Агент способен передавать и принимать сообщения, он реализует функциональность сервиса. Существует различие между **агентом** и **сервисом** - один и тот же сервис может быть обеспечен разными агентами.

Механизм обмена сообщениями определяется в описании сервисов (Web Services Description), которое представляет собой спецификацию интерфейса сервиса и охватывает форматы сообщений, типы данных, транспортные протоколы, способы сериализации, используемые при обмене между агентами заказчика и поставщика услуг. Кроме того, описание сервиса содержит указание на одну или несколько точек в сети (endpoint), откуда доступен поставщик.

Технология **Universal Description, Discovery and Integration (UDDI)** предполагает ведение реестра веб-сервисов. Подключившись к этому реестру, потребитель сможет найти веб-сервисы, которые удовлетворяют его потребностям. Технология UDDI дает возможность поиска и публикации нужного сервиса, как человеком, так и программой-клиентом.

Стек технологий, реализующий архитектуру веб-сервисов, представлен на рисунке 2.

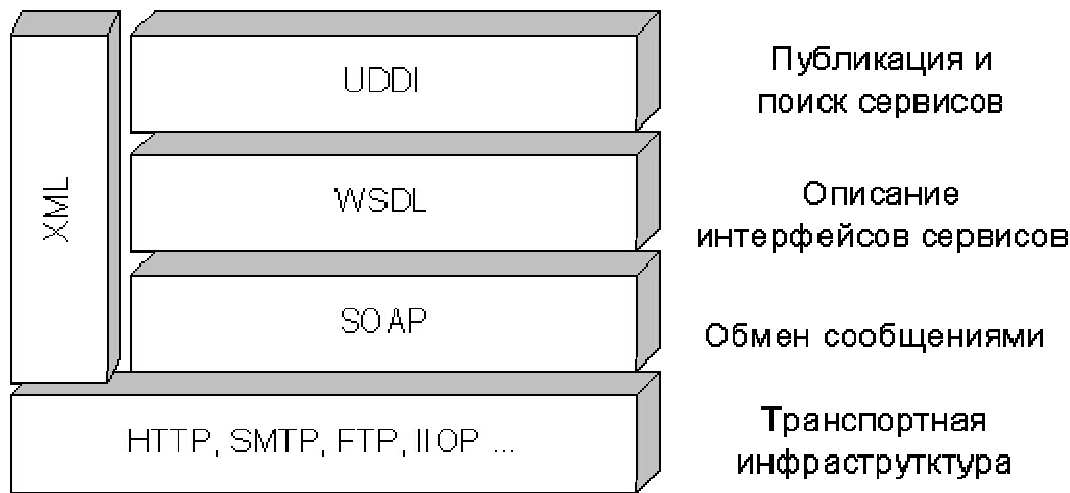


Рисунок 2. Стек технологий архитектуры веб-сервисов

Концепция веб-сервисов подразумевает, что отдельные веб-сервисы обладают определенной ограниченной функциональностью. А для решения более-менее сложных задач требуется использовать функциональность нескольких сервисов. Поэтому в ходе развития архитектуры веб-сервисов возникли понятия "*композиция Web-сервисов*" (Web-services composition) и "*поток Web-сервисов*" (Web-services flow), в последнее время взамен этих понятий используют соответственно "*оркестровка*" (Web Service Choreography) и "*хореография*" (Web Service Choreography) веб-сервисов. Эти понятия отражают взаимодействие сервисов и последовательность их выполнения. Другими словами, приложения, построенные с использованием веб-сервисов, рассматривают как *приложения, основанные на потоках работ* (Workflow-based applications).

При рассмотрении приложений, основанных на потоках работ, выделяют три уровня как это показано на рисунке 3 [4].

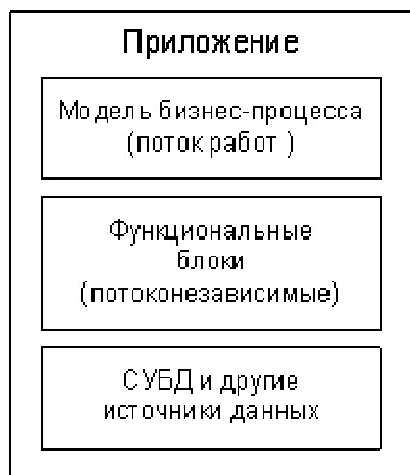


Рисунок 3. Приложение, основанное на потоках работ

Веб-сервисы являются функциональными блоками и соответствуют единицам работ в терминах потоков работ. Способы нахождения, подключения, использования единиц работы (т.е., веб-сервисов) обеспечиваются стеком технологий, который был описан выше и представлен на рисунке 2.

В настоящий момент актуальной проблемой является разработка стандартизованного способа организации приложения на уровне модели бизнес-процесса, а также связанная с

ней проблема методик аналізу, розробки, моніторингу бізнес-процесів, використовуючих в своїй роботі веб-сервіси, або навіть побудованих переважно на веб-сервісах.

Веб-сервіси широко використовуються для інтеграції, в тому числі для взаємодії бізнес-процесів компаній-партнерів. Це підвищує важливість механізму організації веб-сервісів.

Для опису систем і додатків, побудованих за архітектурою веб-сервісів, на рівні моделі бізнес-процесу ведучими ІТ-компаніями пропонувалися різні проекти стандартів: Wf-XML (від Workflow Management Coalition), WSFL (IBM Web Services Flow Language), XLANG (Microsoft's XLANG: Business modeling language for BizTalk), PIPs (RosettaNet's Partner Interface Process), а також деякі інші.

К нинішньому моменту найбільший вага мають **BPEL4WS (Business Process Execution Language for Web Services)**, підготовлений IBM, Microsoft і BEA Systems, і **WSCI (Web Service Choreography Interface)** корпорації Sun Microsystems. Обидва ці проекти були прийняті в організації, стандартизуючі технології архітектури веб-сервісів:

- WSCI з 2002 року розвивається робочою групою консорціуму W3C (організована робоча група Web Services Choreography Working Group) [7];
- для розвитку BPEL4WS в 2003 році в консорціумі OASIS був створений технічний комітет - OASIS Web Services Business Process Execution Language TC (WS-BPEL TC) [6].

WSCI відображає концепцію хореографії сервісів. BPEL4WS призначений для реалізації оркестровки сервісів.

Стандарти хореографії і оркестровки ґрунтуються на стандартизованому опису веб-сервісів - WSDL. Веб-сервіси, що беруть участь в взаємодії, представлені своїми інтерфейсами.

Архітектура веб-сервісів (або більш загальна концепція - архітектура, орієнтована на сервіси) є динамічно розвиваючимся напрямком в інформаційних системах і ПО. Існуючий набір галузевих стандартів уже сьогодні дозволяє вирішувати завдання інтеграції (в тому числі, навіть на рівні бізнес-процесів).

## **Типи архітектури Web 2.0 технологія RIA. сервісно орієнтована архітектура SOA**

Багатьом з вас відомі деякі слова, які стосуються цих технологій, ці технології зараз використовує майже кожен потужний ресурс, слова Adobe Flash, Adobe Media Player, Microsoft Silverlight, також AJAX більшості користувачів відоме їх значення.

Власне в сьогоднішній публікації я і пропоную розібратися для чого використовують ці технології, як вони працюють і взаємодіють на одному ресурсі, що підрозумівається під технологією RIA, що означає термін SOA, які терміни включає в себе соціальна мережа, про все це буде йти мова в сьогоднішній публікації.

Розпочну публікацію з такого поняття як технологія RIA.

Технологія RIA (Rich Internet Application) якщо дослівно перекласти з англійської, то буде означати "багаті інтернет додатки", містить такі відомі додатки як Ajax, Flash, дозволяють створювати складні веб додатки, з метою розмивання відмінностей в інтерфейсах, і часу відповідей на звичайні настільні додатки, які поставляються з допомогою плагінів власних і незалежних веб браузерів, через тестування механізмів, або через віртуальні машини.

Термін "Rich Internet Application" (RIA), означає (розширення інтернет додатків) був введений в 1990 році компанією Macromedia, і спочатку підрозумівав розширенню

інтерфейсів додатків, медіа контенту і випадки вирішення узагальнених рішень, шляхом введення власних розширень. Веб додатки збагатились з появою технології RIA, вони стали більш складними, але заодно їх стало складніше реалізовувати.

RIA використовує настільні додатки наряду з веб додатками. З допомогою цієї технології можна створювати такі складні елементи як “Он лайн ігри”, вони також являються напрямком де додаток RIA дуже важливий. Для запуску RIA необхідний підходящий шаблон, додаток з інтернет розширенням, який необхідно встановити з допомогою операційної системи компютера, перед запуском додатку.

Шаблон програмного забезпечення як правило відповідає, за завантаження, оновлення, перевірку і приведення в виконання RIA. Приклади RIA Adobe Flash, Adobe Air, Java/JavaFX, Microsoft Silverlight

Технологія Ajax (Asynchronous JavaScript і XML) являється новою моделлю для розширення виконання веб – додатків, шляхом нового об'єднання існуючих технологій. Ідея Ajax складається в використанні об'єктів JavaScript типу XMLHttpRequest для асинхронного виклику веб сервера. Технологія Ajax являється даним стандартом, більшість браузерів підтримує XMLHttpRequest.

Ajax (від англ (Asynchronous JavaScript und XML) “асинхронний JavaScript і XML” підхід до побудови інтерактивних користувацьких інтерфейсів, веб додатків, заключається в фоновому обміну даних браузера, з веб сервером. В результаті при оновленні даних веб сторінка не перезавантажується повністю, і веб додатки стають більш швидкими і зручними.

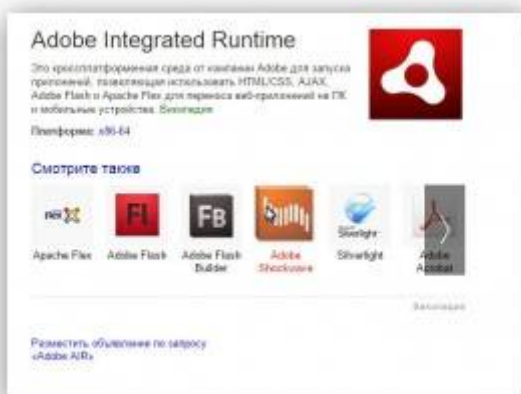
Платформа технологій Adobe для RIA (Rich Internet Application) була розширена шляхом розробки такого програмного забезпечення, як Adobe AIR і Adobe Flex, яке використовується для швидкого створення інтерактивного контенту і додатків, яка здолало різницю між браузерами, робочими столами і операційними системами.

Ведучі компанії в таких галузях як: медіаіндустрія, розваги, фінанси, і соціальні мережі (наприклад Deutsche bank, The New York Time Company ) адаптували технології Adobe AIR, в цілях забезпечення інтересної і ефективною взаємодії з клієнтами.

Технології Adobe AIR включають інструменти, серверні платформи, послуги і виконуючі файли, які діють разом, створюючи тим самим досвід широкого спектру дій.

Adobe AIR (Adobe Integrated Runtime) — дозволяє розробникам створювати RIA на робочому столі, використовуючи техніку і технології веб (такі як HTML, AJAX, PDF, Adobe Flex, Adobe Flash ), які вони вже використовують.

Додатки основані на Adobe AIR, володіють перевагами додатків RIA із браузера (швидкість, просте використання і доступ практично з любого місця), так і перевагами настільних додатків, здатністю писати і читати локальні файли, працювати з іншими додатками на компютері користувача і зберігати дані, які зберігаються на робочому столі.



Додатки основані на AIR можуть працювати без інтернету, а потім активувати функції і відправляти дані коли інтернет з'єднання встановлено. За приклад можна взяти eBay Desktop, який дозволяє продавцям створювати off-line список і потім викладати його на eBay коли інтернет підключений. Доступно більше 180 додатків, які використовують Adobe AIR.

Adobe Flex призначена для створення інтерактивних додатків RIA. Вона поєднує в собі формат MXML (Macromedia XML), з мовою скриптів Actionscript. Кінцевий результат буде у виді файлів Flash (SWF), який буде містити інтерактивні елементи інтерфейсу. Framework Flash надається компанією Adobe з відкритим кодом.

Для ефективного створення додатків можна використовувати Adobe Flex Builder, пропонується з безкоштовною ліцензією для використання навчання в домені, або Adobe Flex Builder Professional на комерційній основі.

Перевага технології Flex складається в тому щоб керувати додатками як у веб браузері, так і настільними додатками з використанням Adobe AIR.

Adobe Media Player являє собою також нові додатки, створені з Adobe Flex (SWF) і оснований на Adobe AIR. Головне призначення програми це відтворювати ролики формату Adobe Flash, програма також підтримує онлайн і офлайн режим, а також потокове відтворення.

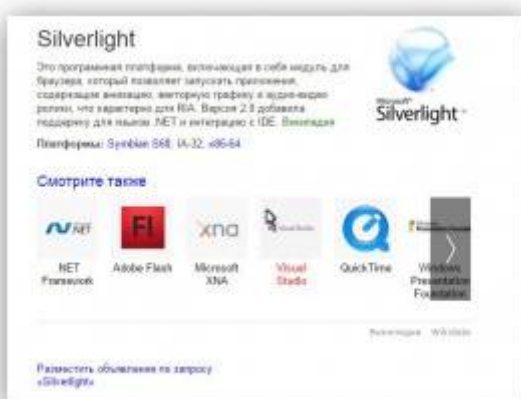
Adobe Media Player являється поєднанням телебачання і інтернету, що дозволяє користувачам відслідковувати свої любими шоу передачі в будь-якому місці, де б вони не знаходились, в будь-який час.

Adobe Media Player був випущений в квітні дві тисячі восьмого року, в ньому вдосконалені технології Flash, з його допомогою компанія Adobe застосовує перегляд навчальних відеороликів по своїм продуктам. Демонструється навчальний відео-ролик, як працювати з тією чи іншою програмою створеною Adobe.

Переваги Adobe Media Player в тому що з його допомогою можна передивлятись відео файли з розширенням SWF, без необхідності використання браузера, також присутня можливість перегляду файлів з розширенням SWF в повноекранному режимі. Програваач також здатний автоматично відстежувати і завантажувати нові серії ваших любимох телевізійних передач, у вказаний час, в результаті чого ви зможете їх передивлятись і в офлайн режимі.

Adobe Media Player — розповсюджується безкоштовно і працює як на операційних системах Windows, так і на Mac OS.

Microsoft Silverlight це програмна платформа яка дозволяє відображати мультимедійний контент в браузері, такі додатки які містять анімацію, векторну графіку, і аудіо відео ролики. Основана Microsoft Net Framework, Silverlight дозволяє дизайнерам і розробникам з легкістю використовувати діючі інструменти для введення RIA (Rich Internet Application) в веб. Для цього дизайнери використовують Microsoft Expression Studio а розробники Visual Studio.



Silverlight може включати мову розмітки додатків XAML (Extensible Application Markup Language) який генерується динамічно на сервері і може також отримати доступ до баз даних, що дозволяє реалізувати складні додатки. Microsoft Silverlight, технологія яка створює конкуренцію Adobe Flash. Дана програма працює у вигляді плагіна, разом з популярними браузерами.

Curl об'єктно-орієнтована мова програмування, враховуючи що Adobe пропонує AIR а Microsoft — Silverlight в якості рішення розвитку RIA, компанія Curl Inc представляє нову об'єктно орієнтовану мову програмування. Вона використовується для створення RIA —



Curl, яка являє собою поєднання мов розмітки, наприклад HTML, скриптів JavaScript, і класичних об'єктно-орієнтованих мов C++ і Java. Curl це більш потужніша мова, чим інші мови програмування які використовуються в технологіях RIA.

Curl рефлексивна об'єктно-орієнтована мова програмування, створена для написання інтерактивних веб-додатків. Потужність Curl забезпечується можливістю комбінувати коди і елементи розмітки, визначати типи параметризації, створювати компіляції і виконувати код в момент виконання програми. На даний час Curl може генерувати більш швидкий код, чим інші мови які використовують в RIA, що особливо важливо при аналізі великих об'ємів даних, або великих форм анімації.

SOA — Service oriented architecture включає FID RSS веб служби Mash-up

Під SOA розуміється підхід архітектури до розробки програмного забезпечення, яке визначає використання послуг, забезпечуючих функціональність, необхідну користувачам. SOA вимагає, щоб нові послуги створювались із існуючих на основі взаємодії.

Як правило ми можемо реалізовувати веб служби з використанням GCI-скриптів, або різних серверних додатків, двома способами.

Функціонально (типу запит відповідь) користувач відвідує сторінку і формулює запит, який проходить через гіперпосилання або через форму.

Умовно (типу запит відповідь) користувач може сформулювати запит про наявність питань до конкретного питання для повернення більш точної відповіді.

SOA може полегшити інтеграцію різних умов в безліч організаціях, сприяти співробітництву і обміну інформації між організаціями, і зовнішніми партнерами. SOA на основі бізнес процесів забезпечує можливість підтримання бізнес моделі, яка виходить за рамки організації, покращує співпрацю, сприяє повному бізнеспроцесу, і підвищує операційну ефективність.

Основна перевага SOA, являється можливість спростити бізнес процеси, що в свою чергу сприяє гнучкому керуванню ними. SOA являє собою можливість зробити бізнес процеси більш прозорими, так що вони можуть бути налаштовані і оптимізовані для більш повного задоволення потреб клієнтів. Час відповіді зменшується, і в той же час зберігається висока якість і надійність.

І саме головне то що SOA зберігає складність інтеграції application-to-application і business-to-business, що значно скорочує затрати, і підвищується технологічний рівень бізнесу.

Соціальна мережа, ще один тип архітектури Web 2.0 включає такі терміни як Tagging, Wiki, Podcast, Blogging

Tagging — визначення мітки в виді ключового слова для організації, швидкого пошуку всередині структури. З допомогою тегів можна досить зрозуміло класифікувати пости, і використовувати їх при оптимізації. За допомогою цієї функції користувач зможе наприклад передивитись на карті місце, де було зроблене фото, чи відзнято відео.

Wiki — веб додаток розробкою структури і наповненням займаються користувачі, вони можуть самастійно створювати і редагувати записи за допомогою тих інструментів які знаходяться на ресурсі Wiki. Термін Wiki використовується і для характеристики спільного програмного забезпечення, яке використовується для створення таких сайтів. Самий відомий тип Wiki це Wikipedia.

Podcasting — спосіб розповсюдження мультимедійних файлів (як правило аудіо і відео) шляхом надання доступу до матеріалів сайту, роблячи їх доступними для повторного використання, формати вмісту RSS і ATOM. Файли можуть бути завантажені і відтворені на компютерах і телефонах, які підтримують формат на якому вони були створені.

Автора підкасту як за звичай називають "podcaster". Підкастинг сайти можуть представляти файли для завантаження і прослуховування он лайн, або офлайн, або для прямого відтворення. Основний метод це метод завантаження через RSS або ATOM.

Blogging — процес ведення блогів, така можливість присутня на багатьох потужних ресурсах, де користувач зможе завести власний блог на якусь тематику, блоги можна завести власний, як правило блог веде автор публікацій, колективний, блог ведеться цілою групою осіб, також і корпоративний, співробітники якоїсь організації ведуть блог.

Звичайно на таких ресурсах присутні правила, які визначені адміністраторами і модераторами, і які потрібно дотримуватись.

**Висновок**

Власне в чому перевага технології RIA, передача клієнту, точніше в ваш браузер необхідну частину інтерфейсу, а більшу частину залишає на сервері, це швидкість, запускається в браузері і не потребує додаткового встановлення, а це зручно, і також запускається локально “в пісочниці” і це безпечно.

Що входить в дану технологію, понад сто вісімдесят додатків, якими можна користуватись, технологія Ajax, в чому її перевага, в швидкості, як працює дана технологія, користувач заходить на якусь сторінку і нажимає на потрібний елемент, браузер відправляє відповідний запит на сервер, а сервер віддає тільки ту частину документа яка змінилась.

Adobe AIR від компанії Adobe потрібна для запуску додатків, завдяки чому появилась можливість використовувати HTML, AJAX, Adobe Flash і Apache Flex для перенесення веб-додатків на ПК і мобільні пристрої.

Adobe AIR – це засіб для розробки інтерактивних додатків від Adobe, можливість розробляти кросплатформені додатки з використанням HTML, Ajax, Flex, Flash

Silverlight — це програмна платформа, що включає в себе модуль для браузера, який дозволяє запускати додатки, що містять анімацію, векторну графіку і аудіо-відео ролики.


Крім компаній Adobe і Microsoft які збагачують своїми додатками технологією RIA, також слід відзначити компанію Curl Inc яка презентує нову об'єктно – орієнтовану мову програмування.

Curl – може бути і мовою розмітки як HTML, мається на увазі звичайний текст показується саме як текст, але в той же час Curl включає в себе можливості об'єктно-орієнтованої мови програмування.

Curl володіє можливістю незалежної роботи аплетів, тобто веб-аплети які виконуються на робочому столі користувача, не залежать від вікна браузера, як це зроблено в Silverlight і в Adobe AIR.

## Web-службы вчера, сегодня, завтра

За сравнительно недолгую историю своего существования Web-службы прошли путь от относительно простого механизма межплатформенного удаленного вызова процедур по протоколу HTTP до ядра новой архитектуры создания приложений — архитектуры, ориентированной на службы (Services Oriented Architecture, SOA).

 Начав со стандартов, которые теперь принято относить к стандартам Web-служб первого поколения — SOAP (Simple Object Access Protocol), WSDL (Web Service Description Language), UDDI (Universal Description, Discovery and Integration Protocol), Web-службы дополнились и собственной архитектурой (Web Services Architecture, WSA) и обширным набором дополнительных стандартов и протоколов для решения таких задач, как гарантированная передача сообщений, обеспечение адресации, безопасность, передача вложенных данных и т.п. (краткое описание существующих сегодня стандартов см. в разделе «Основные стандарты Web-служб»).

Примитивные средства создания Web-служб превратились в мощные программные продукты, позволяющие визуально создавать и потреблять Web-службы, локально или удаленно управлять ими, выполнять мониторинг их активности и производительности.

Вопросы, связанные с обеспечением полной совместимости Web-служб, реализованных на различных платформах, выделились в особую категорию. Появилась даже специальная организация — Web Services-Interoperability Organization (WS-I), отвечающая за создание

профилей, описывающих принципы совместимых Web-служб. В настоящее время уже разработан базовый профиль (WS-I Basic Profile) для SOAP 1.1 и WSDL 1.0.

В данной публикации мы кратко рассмотрим базовые стандарты и технологии, используемые для создания Web-служб, и ознакомимся с основными стандартами Web-служб, разработанными в настоящее время.

## Основы Web-служб

**В** основе Web-служб лежат несколько простых принципов. Возможные для вызова команды описываются на языке WSDL; непосредственная активизация команд происходит в виде посылки SOAP-сообщений по адресу, где располагается Web-служба (используется стандартный протокол HTTP); для поиска Web-служб существуют глобальные или локальные (внутренние) каталоги, поддерживающие стандартные службы обнаружения UDDI. Не вдаваясь в технические подробности, можно отметить, что все современные средства разработки ведущих производителей поддерживают создание Web-служб, а программные платформы (будь то серверные операционные системы или серверы приложений) обеспечивают выполнение Web-служб.

В Web-службах везде используется язык XML. Он служит, в частности, для описания сообщений, которыми могут обмениваться Web-службы и их потребители. SOAP-сообщение — это XML-документ, состоящий из трех базовых элементов: <Envelope>, <Header> и <Body>. Язык WSDL базируется на языке XML и позволяет создавать XML-документы, описывающие методы Web-служб, параметры методов, способы их вызова и т.п. Для того чтобы воспользоваться специализированными Web-службами в рамках механизмов обнаружения UDDI, следует составить SOAP-сообщения и интерпретировать возвращаемые XML-документы.

Жизненный цикл Web-службы можно условно разделить на три фазы: первая — программирование и публикация, вторая — поиск в каталоге, третья — потребление из клиентского приложения (рис. 1).



Рис. 1. Жизненный цикл Web-службы

Поддержка на уровне средств разработки и программных платформ, относительная простота создания и использования Web-служб и практически повсеместная доступность Интернета привели к тому, что за короткий срок появилось множество Web-служб, позволяющих

получать различные данные — от прогнозов погоды и транспортных расписаний рейсов до курсов валют, котировок акций и даже гороскопов.

Некоторые примеры Web-служб представлены в табл. 1.

Назначение Web-службы	Адрес
Web-службы для добавления в базу данных заявок на перевозку груза и предложений транспорта. В виде Web-служб также реализован доступ к транспортным новостям сайта, тематика которых охватывает все виды перевозок и сопутствующие области — таможенно, страхование и т.д.	<a href="http://www.perevozki.ru/">http://www.perevozki.ru/</a>
Web-службы для получения оперативной информации об актуальном расписании, прилете/вылете самолетов, состоянии рейсов. Сервис «Справка о рейсе/Табло аэропортов» предоставляет следующий набор функций: <ul style="list-style-type: none"> <li>- AirportList — список аэропортов;</li> <li>- AirportInfo — информация об аэропорте по его коду;</li> <li>- DateList — список дат, по которым имеется информация;</li> <li>- Arrival/Departure — информация о прибытии/отправлении рейсов;</li> <li>- FlightSearch — поиск рейса по номеру;</li> <li>- FlightInfo — подробная информация о текущем состоянии рейса (по ключу, полученному в результате вызова Arrival/Departure/FlightSearch).</li> </ul> Сервис «Расписание рейсов» предоставляет следующий набор функций: <ul style="list-style-type: none"> <li>- AirportAlphabetList — список букв, на которые начинаются названия аэропортов;</li> <li>- AirportList — список аэропортов на указанную букву;</li> <li>- AirportInfo — информация об аэропорте;</li> <li>- Search — поиск рейсов;</li> <li>- Calendar — информация о других датах, в которые летает указанный рейс</li> </ul>	<a href="http://webservices.aeroflot.ru/">http://webservices.aeroflot.ru/</a>
Использование Web-служб для извлечения информации из поисковой системы Google позволяет разработчикам выполнять запросы более чем к 8 млрд. Web-страниц. (Справедливости ради отметим, что Yandex предоставляет похожую услугу, но базирующуюся на XML-запросах, посылаемых методом POST или методом GET (протокол HTTP) без использования SOAP-запроса.)	<a href="http://www.google.com/apis/">http://www.google.com/apis/</a>
Создание набора Web-служб — Amazon Web Services (AWS) — превратило онлайн-магазин Amazon в полноценную платформу для электронной коммерции. В настоящее время в рамках Amazon Web Services доступны следующие службы: <ul style="list-style-type: none"> <li>- Amazon E-Commerce Service (ECS);</li> <li>- Amazon Simple Queue Service;</li> <li>- Alexa Web InformationService</li> </ul>	<a href="http://www.amazon.com/">http://www.amazon.com/</a>

Таблица 1. Примеры Web-служб

Следует отметить, что, возможно, самой важной особенностью Web-служб является их независимость от платформы. Это означает, что Web-служба и ее потребитель могут быть реализованы практически на любой программной платформе, причем как сама служба, так и ее потребители могут быть реализованы на разных платформах — минимальным требованием к потребителям является поддержка протокола HTTP и возможность программной обработки XML-документов.

По мере возрастания интереса к Web-службам появлялось все больше реализаций Web-служб для решения корпоративных задач; прежде всего Web-службы применялись для интеграции приложений, для обеспечения создания функциональных модулей в гетерогенных средах, для реализации уровней абстракции бизнес-логики от клиентских приложений и т.п.

Применение Web-служб в качестве основной технологии для создания корпоративных приложений привело к тому, что базовые протоколы и стандарты, удовлетворявшие требованиям разработчиков публичных Web-служб, оказались не слишком пригодными для решения задач, стоящих перед корпоративными программистами. В результате появилась

масса дополнений и расширений базовых протоколов, ориентированных на растущие потребности к повсеместному использованию Web-служб.

## Основные стандарты Web-служб

В табл. 2 собрана информация об основных стандартах, связанных с Web-службами. Стандарты и протоколы сгруппированы по базовым функциональным группам — передача сообщений, безопасность, надежность, транзакции, метаданные и бизнес-процессы. Получить полное описание вышеприведенных спецификаций можно на сайте Microsoft по адресу <http://msdn.microsoft.com/webservices/understanding/specs/default.aspx>.



Рис. 2. Распределение стандартов Web-служб по уровням

Передача сообщений (Messaging)	
SOAP	Протокол для обмена структурированной информацией в распределенных средах. Стандарт опубликован W3C
WS-Addressing	Описывает механизмы адресации Web-служб, нейтральные к транспортному протоколу. Разработан компаниями IBM, BEA и Microsoft
WS-Eventing	Описывает протокол, по которому одна Web-служба может проявить интерес к получению сообщений о возникающих в другой Web-службе событиях. Разработан компаниями BEA, Tibco и Microsoft
Безопасность (Security)	
WS-Security	Описывает расширения протокола SOAP для обеспечения целостности и конфиденциальности сообщений. Является частью стандарта OASIS Web Services Security 1.0
WS-Trust	Расширяет стандарт WS-Security, добавляя к нему возможность запроса и получения security tokens.
WS-SecureConversation	Описывает язык Web Services Secure Conversation Language, который может использоваться поверх протоколов WS-Security и WS-Trust для обеспечения безопасных коммуникаций между службами. Разработан компаниями IBM, RSA, Verisign и Microsoft
WS-Federation	Язык Web Services Federation Language используется для обеспечения механизмов передачи аутентификационных и авторизационных данных между различными доверительными доменами. Разработан компаниями IBM, BEA, RSA, Verisign и Microsoft
Надежность (Reliability)	
WS-ReliableMessaging	Описывает протокол, позволяющий осуществлять надежную доставку сообщений между распределенными приложениями. Разработан компаниями IBM, BEA, TIBCO Software и Microsoft (альтернативный протокол был предложен компаниями Sun Microsystems, Oracle, Fujitsu, NEC и Hitachi под названием WS-Reliability)
Транзакции (Transactions)	
WS-Coordination	Описывает протокол для координации действий распределенных приложений. Разработан компаниями IBM, BEA и Microsoft
WS-AtomicTransaction	Предоставляет определение координации атомарных транзакций в рамках протокола WS-Coordination. Разработан компаниями IBM, BEA и Microsoft
WS-BusinessActivity	Предоставляет определение координации бизнес-активностей в рамках протокола WS-Coordination. Разработан компаниями IBM, BEA и Microsoft
Метаданные (Meta Data)	
WSDL	Язык описания Web-служб, основанный на XML. Содержит абстрактное описание операций и сообщений для вызова этих операций, а также их конкретное описание для различных сетевых протоколов и форматов сообщений. Опубликован W3C
UDDI	Спецификация описывает основанные на SOAP Web-службы для программного обнаружения Web-служб. Опубликована OASIS
WS-Policy	Задает основанный на языке XML синтаксис для описания политик Web-служб: требований, возможностей и т.п. Разработан компаниями IBM, BEA, SAP и Microsoft. Расширения — WS-PolicyAttachment, WS-PolicyAssertions, WS-SecurityPolicy и др.
WS-SecurityPolicy	Расширяет спецификацию WS-Security, объясняя, каким образом описывать требования к безопасности Web-служб на уровне политик
WS-MetadataExchange	Определяет три пары сообщений, которые могут использоваться для получения трех основных типов метаданных: для получения политик уровня WS-Policy, для получения описания службы на языке WSDL и для получения XML-схемы для указанного пространства имен. Разработан компаниями IBM, BEA, SAP и Microsoft
WS-Discovery	Описывает протокол для обнаружения Web-служб. Расширяет функциональность UDDI за счет поддержки устройств и систем, которые не всегда подключены к сети. Разработан компаниями Microsoft, BEA, Canon и Intel
Бизнес-процессы (Business Process)	
BPXL4WS	BPXL4WS (Business Process Execution Language for Web Services) — это язык для описания того, как отдельные Web-службы могут быть объединены для создания комплексных, надежных бизнес-приложений. Служит для описания бизнес-процессов, способных потреблять и предоставлять Web-службы. Данная спецификация является дальнейшим развитием языков Microsoft XLANG и IBM WSPF. Разработана компаниями Microsoft, IBM, BEA при участии SAP и Siebel. Дальнейшая работа над стандартом проводится в рамках комитета OASIS WSPFL Technical Committee

Таблица 2. Распределение стандартов Web-служб по уровням

Отметим, что различные компании по-разному обеспечивают поддержку стандартов в своих продуктах, средствах разработки и программных платформах. Например, компания Microsoft выпускает бесплатные программные компоненты под названием Web Services Enhancements, которые доступны для загрузки с Web-сайта компании, расположенного по адресу <http://msdn.microsoft.com/webservices/downloads/default.aspx>.

## Вместо заключения

Web-службы все активнее проникают в область, уже занятую объектно-ориентированными и компонентными технологиями, представляя собой альтернативу традиционным подходам к созданию корпоративных приложений. В ближайшие годы можно ожидать серьезных инвестиций в технологии, связанные с Web-службами, развертываемыми и потребляемыми внутри компаний (согласно исследованиям IDC, к 2008 году объем рынка Web-служб превысит 11 млрд. долл.).

К тому же следует ожидать дальнейшего развития отечественных Web-служб (как бесплатных, так и коммерческих) и появления комплекса других служб, связанных с данной технологией. Возможно, это произойдет в рамках проекта «Электронная Россия», а возможно — по инициативе некоторых крупных поставщиков услуг. Впрочем, каким образом это будет происходить, не так уж и важно — важно то, что Web-службы уже пришли в Россию, и технологии их создания и потребления готовы к использованию.

## Литература

1. Jonathan Sapis, Will Web services and SOA change the development world? www.TechRepublic.com, August 2003.
2. Michael Pallos, Service-oriented architecture: A Primer. EAI Journal, December 2001.
3. Web Services Glossary, <http://www.w3.org/TR/2003/WD-ws-gloss-20030808>
4. F. Leymann, D. Roller, "Workflow-based applications" // IBM Systems Journal. 1997, vol.36
5. Web Services Architecture. <http://www.w3.org/TR/ws-arch/>
6. OASIS Web Services Business Process Execution Language TC - [http://www.oasis-open.org/committees/tc\\_home.php?wg\\_abbrev=wsbpel](http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsbpel)
7. Web Services Choreography Working Group. - <http://www.w3.org/2002/ws/chor/>