

УДК 681.3

ПРИНЯТИЕ РЕШЕНИЙ И ДЕЛОВАЯ ИНТЕГРАЦИЯ В ВЕБ

К.И. Прокопенко, А.И. Прокопенко

Харьковский национальный университет радиоэлектроники

e-mail: kyrylo.prokopenko@gmail.com

Современный подход к обмену информацией между различными корпоративными информационными системами основывается на использовании стандарта XML и Web-сервисов. Web-сервисы базируются на XML-стандарте и позволяют компаниям интегрировать свои внутренние информационные системы, а также обеспечивать взаимодействие с внешними партнерами. Отличие технологии Web-сервисов от предыдущих технологий интеграции в том, что эта технология позволяет решить задачу легче, быстрее и с меньшими затратами. При этом сохраняя уже сделанные инвестиции в существующие компьютерные информационные системы и контроль над ними.

Web-сервисы представляют путь интеграции бизнес-функций, выполняющихся в разных приложениях, в рамках одного “е-бизнес-процесса”, т.е. бизнес-процесса, выполняющегося через Интернет.

Технически Web-сервисы – это модульные бизнес-приложения, имеющие стандартные интерфейсы для работы через Интернет, другими словами, являются слабосвязанными компонентами ПО, доступными для использования через стандартные интернет-протоколы. Таким образом, Web-сервис – это сервис, к которому клиент может получить доступ и использовать через сеть Интернет. При этом сервис действует как черный ящик: клиент должен только найти этот сервис, знать, какую функциональность он представляет, а детали, как данный сервис работает, не важны для клиента. Он только вызывает сервис и обменивается информацией, используя стандартный формат обмена сообщениями.

Несомненно, будущее за распределенными приложениями. В последнее время существенно возросли требования в сложных интеллектуальных расчетах, практически каждое нововведение, связанное с удобством для использования конечным пользователем так или иначе связано с интеллектуальным анализом, поскольку именно это делает его легким и интуитивно понятным пользователю, поэтому тема веб-сервисов интеллектуального анализа данных является чрезвычайно актуальной и будет развиваться в ближайшем будущем очень интенсивно.

Сервисно-ориентированная архитектура (Service-Oriented Architecture, SOA) – это компонентная модель, состоящая из отдельных функциональных модулей приложений, называемых сервисами, имеющих определенные согласно некоторым общим правилам интерфейсы и механизм взаимодействия между собой.

Любое приложение с функциональной точки зрения может быть представлено совокупностью функций; ресурс, реализующий функцию, есть сервис. Таким образом, приведенное определение требует для представления и реализации любого приложения в рамках SOA проведения его полной декомпозиции до уровня отдельных функций.

В понятие независимости сервиса вкладывается следующий смысл: сервисы функционируют независимо от других информационных систем, являются функционально самостоятельными объектами. Они представляют собой “черные ящики” для любых внешних приложений: внешние приложения не знают, как сервис формирует из входных данных выходные. Все, что им известно – что необходимо подать на вход сервиса и что следует ожидать на его выходе.

Функция (или функции), которую реализует данный сервис, должна быть однозначно описана согласно определенным, принятым для всех сервисов, правилам. Должен быть описан набор и типы входных данных, а также набор и типы выходных данных.

Для обеспечения взаимодействия между различными сервисами введена необходимость определения интерфейса для веб-сервиса. Для внешних по отношению к сервису информационных систем не должно иметь значения на каком языке программирования реализован веб-сервис, на какой программно-аппаратной платформе он функционирует, локально или удаленно он расположен. Внешняя информационная система должна иметь возможность взаимодействовать с сервисом (т.е. передать ему входные данные и получить выходные) вне зависимости от указанных его особенностей.

Поэтому, переходя в мир абстракций, можно дать следующее, более сильное, определение сервисно-ориентированной архитектуры. Сервисно-ориентированная архитектура – это архитектура приложения, в рамках которой все функции приложения являются независимыми сервисами с четко определенными интерфейсами, которые можно вызывать в нужном порядке с целью формирования бизнес-процессов.

Сервисно-ориентированная архитектура, будучи практической (и практичной) концепцией, должна соответствовать определенным требованиям, предъявляемых к ней современным состоянием бизнес-отношений и информационных технологий, а также тенденциями их совместного развития:

1) обеспечивать преемственность инвестиций в ИТ, сохранение существующих информационных систем и их совместное эффективное использование для повышения ROI от ИТ-вложений;

2) обеспечивать реализацию различных типов интеграции:

- пользовательская интеграция (user integration) – обеспечение взаимодействия информационной системы с конкретным персонифицированным пользователем;

- интеграция приложений (application connectivity) – обеспечение взаимодействия приложений;

- интеграция процессов (process integration) – интеграция бизнес-процессов;

- информационная интеграция (information integration) – интеграция с целью обеспечения доступности информации и данных;

- интеграция новых приложений (build to integrate) – интеграция новых приложений и сервисов в существующие информационные системы.

3) обеспечивать поэтапность внедрения вновь созданных и миграции существующих информационных систем;

4) иметь стандартизованную технологическую обеспеченность реализации и инструментарий разработки, совокупно предоставляющие наилучшие возможности повторного использования приложений, внедрения новых и миграции существующих информационных систем;

5) позволять реализацию различных моделей построения информационных систем, в особенности, таких как порталные решения, grid-системы и on-demand-системы.

Современный уровень развития сервисно-ориентированных архитектур позволяет утверждать, что все указанные требования в той или иной мере выполняются, но дело в том, что в данный момент технологии и сервисы, предоставляемые пользователю стали

настолько объемными и большими, что содержать их в каждом отдельном веб-сайте стало сложно с точки зрения разработки во-первых и с точки зрения поддержания во-вторых. Поэтому стали активно создаваться веб-сервисы, которые могут иметь исключительно администраторскую часть в виде веб-сайта, остальное представлено лишь набором интерфейсов для получения и предоставления данных. Такой подход максимально абстрагирует создателя сайта от решения сложных, ресурсоемких нетривиальных задач, а так же улучшает общую архитектуру приложения, делая ее более прозрачной и понятной.

Как видно из рисунка 1, клиентам веб-сервисов могут быть как отдельно взятые страницы, целые веб-сайта, так и другие веб-сервисы.

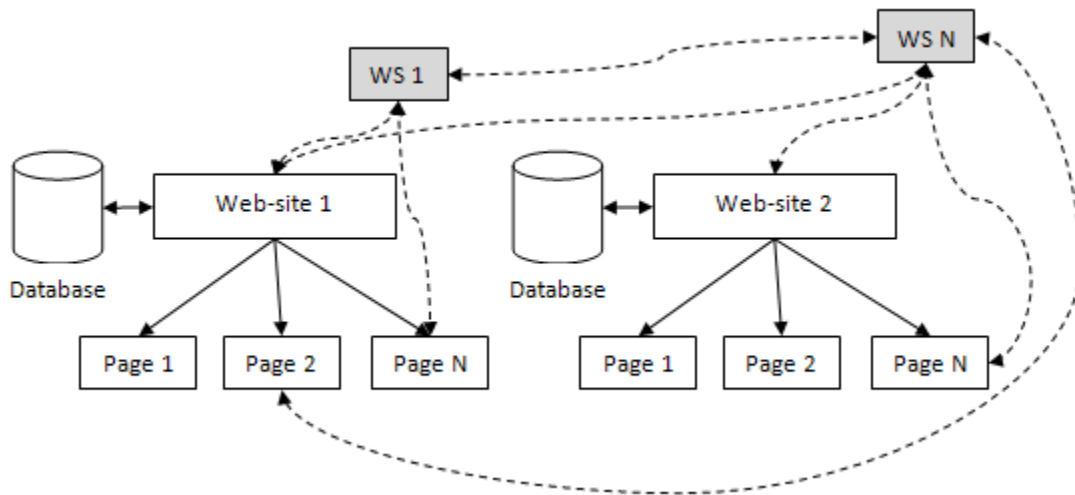


Рисунок 1 – Распределенная веб-сервисная архитектура

Веб-сервис интеллектуального анализа данных представляет собой элемент распределенной сервисной архитектуры, и, описывая общими словами, будет состоять из веб-части, служащей для управления веб-сервисом, и, собственно, из самого веб-сервиса, что будет представлено набором интерфейсов взаимодействия с веб-сервисом (так называемое API, рисунок 2).

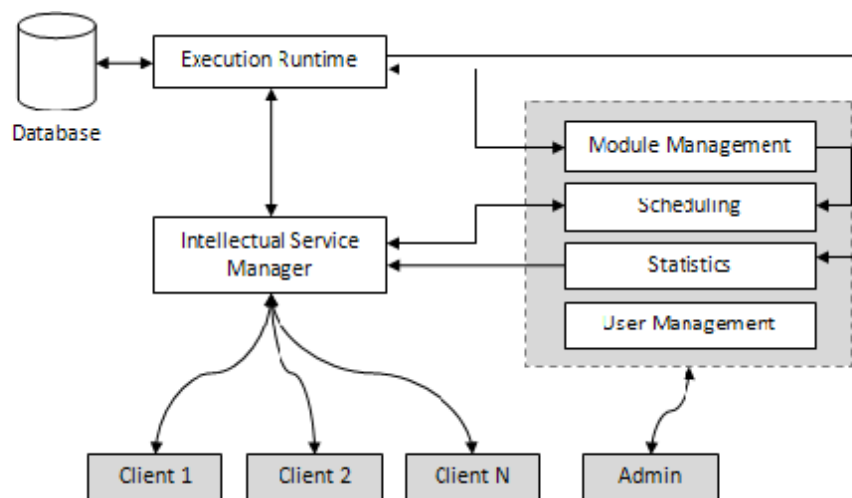


Рисунок 2 – Строение веб-сервиса интеллектуального анализа данных

Внутри веб-сервис кроме административной веб-части и API веб-сервиса, состоит из следующих блоков:

1) Блок интеллектуального управления запросами. Этот блок занимается тем, что обрабатывает запросы, поступающие от API веб-сервиса, проверяет аутентификацию пользователя, сравнивает с информацией в подписках веб-сервиса, оценивает текущую загрузку сервера, и ставит запрос в очередь выполнения запросов. Еще одной немаловажной функцией данного блока является то, что он анализирует результаты статистики и корректирует расписание и выделяемые ресурсы для тех или иных задач на сервере от тех или иных пользователей для обеспечения максимальной производительности;

2) Блок выполнения задач. Занимается тем, что является средой выполнения задач., организует выполнение задач согласно инструкциям блока интеллектуального управления задачами. Также, его задачей является организация и централизованный доступ к базе данных;

3) База данных хранит информацию, такую как обучающие выборки, наборы данных и т.п.

В целом, веб-сервис интеллектуального анализа данных спроектирован таким образом, чтобы представлять собой всего лишь набор модулей, как можно меньше взаимосвязанных и общающихся в свою очередь согласно некоторому интерфейсу.

Такая структура позволяет легко расширять веб-сервис, а также с меньшими усилиями перевести приложение на более распределенный уровень, т.е. можно разнести его основные модули по отдельным серверам, получив огромный прирост в производительности при этом не нарушив их взаимодействие.

Также, данная структура позволяет решить проблему расширения веб-сервиса путем поддержки подключаемых модулей, для чего был описан интерфейс модуля, реализовав который, сторонние разработчики могут сделать функциональность, в полной мере поддерживаемую и таковую, которая может быть использована веб-сервисом интеллектуального анализа данных.

Сам веб-сервис интеллектуального анализа данных представляет собой не что иное, как некоторый вычислительный центр, обладающий огромным набором вычислительных методов и приемов, и позволяющий пользователям выполнять самые разнообразные задачи, связанные с расчетами и анализом.

Список используемой литературы

1. Шелуденков А.В. Разработка и исследование клиент-серверной архитектуры для взаимодействия систем. <http://masters.donntu.edu.ua/2007/fvti/sheludenkov/diss/index.htm>
2. Системы «клиент-сервер» <http://belani.narod.ru/1/Lklser2.htm>
3. Лазарь Б. Клиент/сервер – это серьезно http://www.osp.ru/lan/1995/05/131813/#part_1
4. Савельев А. Технология, которая перевернёт веб <http://www.computerra.ru/hitech/39239>