

Литература:

1. Астафьева Людмила. Корпоративный университет: этажи системы. Журнал "Отдел кадров" (г. Киев), № 19 (130), октябрь 2004 года.
2. Вишнякова Марина. Ловушки обучения. Москва, журнал "Консультант" (№ 9, 2005),

Наливайко Д.В., Карасик А.А., Катков А.Ю., Третьяков В.С.
ПОСТРОЕНИЕ СЕРВИСНО-ОРИЕНТИРОВАННОЙ АРХИТЕКТУРЫ СОВРЕМЕННОЙ
ИНФОРМАЦИОННОЙ СИСТЕМЫ

NalivaykoDen@list.ru

Российский государственный профессионально-педагогический университет
г. Екатеринбург

Большинство современных информационных систем построено на основе традиционной трехуровневой клиент-серверной архитектуры. Однако в ряде случаев в указанную архитектуру могут быть внесены корректировки, оправданные с точки зрения условий применения и развития систем, а также улучшения их характеристик и свойств.

Перечислим основные свойства, которыми должна обладать современная информационная система.

Распределенность. Современная система редко функционирует сама по себе без взаимодействия с другими системами. Кроме того, все чаще возникает необходимость разделения системы на подсистемы, постоянно взаимодействующие между собой. Причем все чаще это взаимодействие осуществляется через сеть Интернет с использованием преимущественно открытых протоколов.

Масштабируемость. Итерационная модель разработки подразумевает реализацию поставленных задач поэтапно, и на начальном этапе часто сложно предсказать полный объем задач и требований, которые необходимо будет выполнить в будущем. Архитектура системы должна обеспечивать беспрепятственное развитие системы в плане увеличения набора предоставляемых сервисов, роста объемов данных и количества пользователей системы без угроз падения производительности и устойчивости системы.

Доступность. Информационная система предоставлять максимальный набор сервисов при минимальном наборе технических ресурсов у ее пользователей. А это значит, что она должна быть ориентирована на работу в условиях низкой пропускной способности информационных каналов (объем передаваемого ввремя работы с системой сетевого трафика между клиентом и сервером должен быть минимален), должна иметь кросс-платформенные клиентские средства, поддерживающие различные операционные системы (или WEB-браузеры).

Модульность. Международный стандарт на разработку программного обеспечения предполагает модульный подход. Он заключается в декомпозиции системы на модули и определении отношений между ними. Разработка каждого модуля ведется независимо (возможно разными группами разработчиков), взаимодействие обеспечивается посредством заранее определенных программных интерфейсов.

Предлагаемая архитектура информационной системы, изображенная на рис. 1, в полной мере отвечает всем перечисленным требованиям.

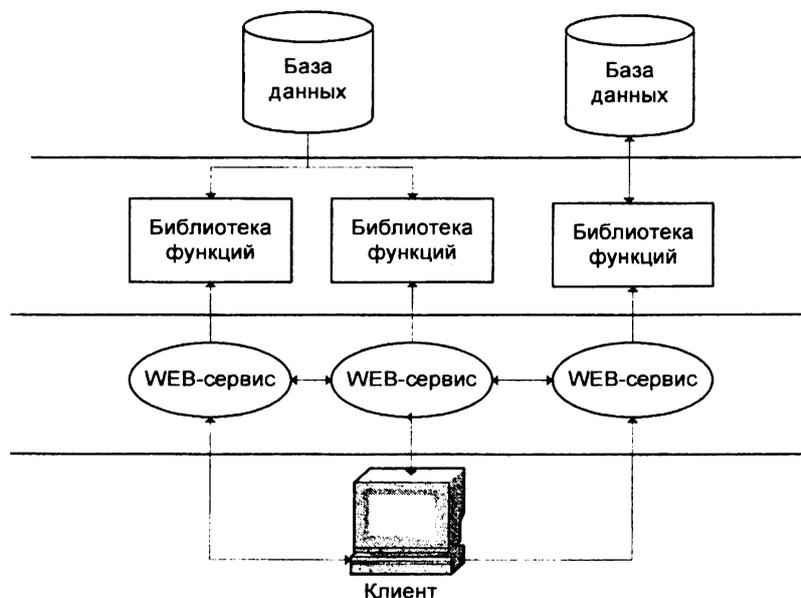


Рис. 1. Предлагаемая архитектура информационной системы

В основе этой архитектуры лежит классическая трехуровневая архитектура клиент-серверного приложения. Здесь присутствуют все её основные элементы, такие как уровень базы данных, уровень бизнес-логики и уровень интерфейса. Особенность заключается в том, что уровень бизнес-логики взаимодействует с уровнем интерфейса (клиентским приложением) посредством web-сервисов.

Активно развивающаяся сейчас технология web-сервисов дает большие преимущества при разработке распределенных приложений. Использование только открытых протоколов и общепризнанных стандартов открывает широкие возможности по передаче информации с использованием web-сервисов через сеть Интернет.

В рамках предлагаемой архитектуры web-сервисы предоставляют доступ к методам объектов бизнес-логики, взаимодействуя с клиентским приложением путем передачи информации в формате XML. Построение интерфейса возлагается полностью на клиентское приложение, даже если оно реализовано в среде браузера.

Такой подход отвечает требованию по минимизации трафика, т.к. в формате XML передается только непосредственно та информация, которая необходима для работы клиентского приложения (в отличие от традиционного web-приложения, где помимо информации передается большой объем сервисных данных, относящихся к внешнему виду страницы).

Существующие технологии позволяют создавать полноценные «богатые» клиентские приложения, взаимодействующие с web-сервисами, как в виде windows-приложений, так и в браузере с использованием AJAX-технологий.

Бизнес-логика приложений, взаимодействие с базой данных и объектное представление предметной области реализованы в библиотеках динамической компоновки. Для реализации своих методов web-сервис использует эти библиотеки либо вызывает методы другого web-сервиса. Клиентское приложение может обращаться за информацией ко множеству web-сервисов, которые могут быть распределены по нескольким серверам. А это означает, что информационная система изначально отвечает требованиям распределенности. Во-вторых, ее разработка может вестись по модульному принципу, когда каждая команда разработчиков отвечает за реализацию определенного набора сервисов. Более того, наличие еще одной программной прослойки, помимо библиотек классов, позволяет вынести на уровень web-сервисов часть общесистемных задач, таких как, например, защиту информации, аутентификацию и авторизацию. Это значительно облегчает процедуру изменения общесистемных механизмов, и позволяет распределить работу по реализации библиотек классов и методов сервисов между разработчиками.

Нужно отметить, что масштабирование информационной системы с предложенной архитектурой также не вызовет затруднений. Во-первых, нагрузка при необходимости может быть легко распределена между неограниченным количеством серверов, каждый из которых возьмет на себя работу с определенным набором объектов бизнес-логики. Во-вторых, использование «богатых» клиентских приложений изначально снизит нагрузку на сервера за счет перенесения уровня интерфейса на машины пользователей системы (по сравнению с традиционным web-приложением).

Таким образом, представленная архитектура информационной системы отвечает всем требованиям, предъявляемым к современным информационным системам. Декомпозиция уровня бизнес-логики в классической трехуровневой архитектуре приложений на библиотеки классов и web-сервисы дает ряд новых преимуществ и возможностей для информационных систем.

Литература:

1. Цикл статей сайта "Технологии веб-сервисов" (<http://www.ubs.ru/ws/ws.html>)
2. Service-Oriented Architecture expands the vision of Web services. Marc Colan (<http://www-128.ibm.com/developerworks/webservices/>)

Поварницын Д.А., Кашеварова Г.Г. РЕАЛИЗАЦИЯ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНОЛОГИИ СОЗДАНИЯ ПРОГРАММНОГО КОДА ДЛЯ РАСЧЕТА НАПРЯЖЕННО-ДЕФОРМИРОВАННОГО СОСТОЯНИЯ ПРОСТРАНСТВЕННЫХ СООРУЖЕНИЙ В ПРОГРАММНОМ КОМПЛЕКСЕ ANSYS

smivt@pstu.ru

Пермский государственный технический университет (ПГТУ)

г. Пермь

В настоящее время специалисты строительной отрасли имеют в своем арсенале большое количество специализированных программных комплексов для проектирования новых зданий и сооружений, таких как LIRA, SCAD, MicroFe и др., в которых реализован метод конечных элементов, а построение геометрической модели осуществляется в интерактивном режиме. При реконструкции или модернизации уже существующих объектов, в аварийных ситуациях с учетом имеющихся дефектов и трещин, на наш взгляд, наиболее рациональным является универсальный программный комплекс ANSYS. Но процесс формирования модели зданий и сооружений в интерактивном режиме в ANSYS является достаточно сложным и требует достаточно много времени для подготовки инженера-строителя. Значительно облегчает работу наличие встроенного в ANSYS языка параметрического проектирования APDL, для создания *программного кода* параметрических прототипов конструкций зданий и сооружений. Многочисленные расчеты зданий и сооружений по программам