

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ
БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

М. А. Артемов,
С. В. Золотарев,
Е. С. Барановский

РАЗРАБОТКА СЕРВИС-ОРИЕНТИРОВАННЫХ ПРИЛОЖЕНИЙ

Учебно-методическое пособие для вузов

Воронеж
Издательский дом ВГУ
2015

Содержание

Введение	6
1. Сервисы.....	7
1.1 Сервис и его роль в SOA	8
1.2 Обобщенная модель разработки ПО на основе SOA	12
1.3 Сервис-ориентированность и ее особенности.....	12
1.4 Мифы и реальность SOA	15
2. Web-сервисы как способ реализации SOA.....	16
2.1 Обмен сообщениями.....	18
Почему SOAP?	18
Структура SOAP сообщения.....	18
Header элемент	20
Body элемент	22
Элемент Fault	24
Транспортировка SOAP сообщения.....	25
2.2 Описание сервиса	27
Что такое WSDL?	27
Структура WSDL	28
Элемент Types	31
Элемент Messages	32
Элемент Binding.....	35
Элемент Ports	36
Элемент Services	37
2.3 Регистрация, поиск и интеграция.....	38
Что такое UDDI?	38
Как работает UDDI?	38

Введение

В последние годы значительный интерес вызывает покомпонентная сборка приложений на базе сервис-ориентированной архитектуры. Существует множество определений термина «сервис-ориентированная архитектура» (SOA, англ. service-oriented architecture). SOA можно рассматривать как модульный подход к разработке программного обеспечения, основанный на использовании распределённых, слабо связанных (англ. loose coupling) заменяемых компонентов, оснащённых стандартизированными интерфейсами для взаимодействия по стандартизированным протоколам [1]. В статье [2] SOA определяется как «a loosely-coupled architecture designed to meet the business needs of the organization» (слабо связанная архитектура, служащая интересам бизнеса).

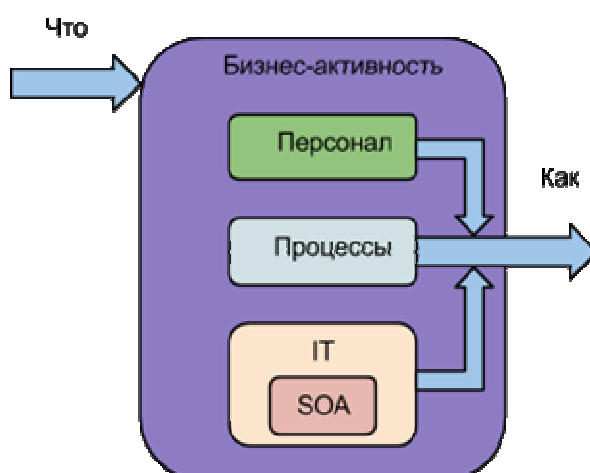
По сути, SOA – это стиль мышления, набор лучших практик и рекомендаций для построения высокоэффективных информационных систем. Разумеется, существуют и конкретные технологии, реализующие принципы SOA.

В данном учебно-методическом пособии рассматриваются вопросы разработки сервис-ориентированных приложений. В разделе 1 обсуждаются сервисы, их свойства, назначение и роль в SOA. Здесь также приводится обобщенная модель разработки программного обеспечения на основе SOA. В разделе 2 рассматриваются Web-сервисы. В разделе 3 дается введение в Windows Communication Foundation (WCF). Раздел 4 посвящен вопросам программирования в WCF.

1. Сервисы

Одним из основополагающих принципов SOA является то, что она должна служить интересам и бизнес-целям предприятия и быть лишь средством повышения эффективности работы, а не рассматриваться как цель сама по себе. Попробуем это пояснить.

В бизнесе важную роль играет набор услуг, которые могут быть предоставлены клиенту. Например, видом услуги может быть доставка определенных товаров клиентам. Для осуществления услуг необходим налаженный процесс, в котором принимают участие люди – персонал предприятия. Для повышения эффективности организации этого процесса можно задействовать широкие возможности программного обеспечения (ПО) и аппаратных средств. SOA помогает оптимизировать работу программного обеспечения и взаимодействия персонала и приносит реальную пользу компании в контексте реализации соответствующей бизнес-активности. Образно говоря, SOA отвечает на вопрос «как делать?», то есть является средством реализации бизнес-активности, которая, в свою очередь, отвечает на вопрос «что делать?», обозначая при этом стратегическую задачу бизнеса.



Слабая связанность составных частей ПО предприятия, внедряющего SOA, обеспечивает его гибкость, то есть способность быстро и успешно реагировать на часто изменяющиеся условия рынка. Для того чтобы понять за счет чего эта низкая связанность достигается, перейдем к рассмотрению основополагающего понятия SOA, имя которому – **сервис**.

1.1 Сервис и его роль в SOA

В [3] дается следующее определение:

SOA – это стиль программирования, архитектурный подход в разработке программного обеспечения, при котором приложение состоит из функциональных модулей с определенным поведением. Эти модули называются сервисами.

В SOA различный функционал приложения может быть доступен через совокупность сервисов. Например, это может быть сервис отправки e-mail или конвертации валюты. Сервисы являются независимыми и часто содержат в себе бизнес-логику, которая может манипулировать релевантными данными. При этом потребитель сервиса не обязан ничего знать о деталях реализации этого функционала. Между потребителем и внутренним устройством самого сервиса пролегает четко обозначенный уровень – публичный интерфейс.

Коммуникации сервиса с клиентом осуществляется с помощью информационных **сообщений (Message)**, структура которых полностью описана так называемой **схемой (Schema)** сервиса. Схема, по сути, является метаданными функционала сервиса, то есть данными о данных. Используя схему, можно корректно сформировать сообщения, которые в последствии будут приняты и прочитаны сервисом для последующего выполнения инструкций, заложенных в сообщении.