Realisation of SOA using Web Services

Adomas Svirskas
Vilnius University
December 2005
Agenda

• SOA Realisation
• Web Services
• Web Services Core Technologies
SOA and Web Services [1]

- SOA is a *way of organising and exposing business functionality* through IT and an *architectural style*, Web Services are *enabling technology standards*

- They are not the same thing:
  - SOA proposes an advancement in the programming model
  - Web Services implement the programming model complete with standards, tools and technologies
SOA and Web Services [1]

• Many existing production SOAs do not primarily use Web services - they are built on Message-Oriented Middleware (MOM)
• Not all deployed Web services-based systems necessarily embrace all the guiding principles of SOA (e.g. loose coupling)
• WS is not the only implementation of SOA, there are some other, e.g. REpresentational State Transfer (REST) [4]
Web Service Definition

• A Web service is a software system designed to support interoperable machine-to-machine interaction over a network
• It has an interface described in a machine-processable format (specifically WSDL)
• Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards

WS – the Very Basics [1]
WS – the Very Basics [2]

Web Service Description Language defines uses Web Service

uses defines

Service Consumer

WSDL Provides only the Technical Description of the Service.
WS – the Very Basics [1]

Service Provider

SOAP Message Handler

Backend or Legacy Application

Service Requester

Internet

SOAP Handler (Proxy)

Service Requester Application

WSDL

Integrated Development Environment (IDE)

WSDL description is generated by inspecting existing application

Code for handling messages is generated from WSDL description
Engaging a Web Service

1. Parties "become known" to each other

2. Agree on semantics & WSD

3. Input Semantics & WSD

4. Interact

Web Services Architecture, W3C, http://www.w3.org/TR/ws-arch/#engaging
Engaging a Web Service [3]

1. The requester and provider entities become known to each other (or at least one becomes known to the other)
2. The requester and provider entities somehow agree on the service description and semantics that will govern the interaction between the requester and provider agents
3. The service description and semantics are realized by the requester and provider agents
4. The requester and provider agents exchange messages, thus performing some task on behalf of the requester and provider entities
Web Services Discovery [3]

• If the requester entity wishes to initiate an interaction with a provider entity and it does not already know what provider agent it wishes to engage, then the requester entity may need to "discover" a suitable candidate.

• Discovery is "the act of locating a machine-processable description of a Web service that may have been previously unknown and that meets certain functional criteria." [WS Glossary]

• The goal is to find an appropriate Web service.

Web Services Architecture, W3C, http://www.w3.org/TR/ws-arch/#wsdisc
Web Service Discovery

Web Services Architecture, W3C, http://www.w3.org/TR/ws-arch/#wsdisc
1. The requester and provider entities "become known to each other":
   a) The discovery service somehow obtains both the Web service description ("WSD") and an associated functional description ("FD") of the service. The functional description is a machine-processable description of the functionality (or partial semantics) of the service that the provider entity is offering. It could be as simple as a few words of metadata or a URI, or it may be more complex, such as a TModel (in UDDI) or a collection of RDF, DAML-S or OWL-S statements.
   
b) The requester entity supplies criteria to the discovery service to select a Web service description based on its associated functional description, capabilities and potentially other characteristics. One might locate a service having certain desired functionality or semantics; however, it may be possible to specify "non-functional" criteria related to the provider agent, such as the name of the provider entity, performance or reliability criteria, or criteria related to the provider entity, such as the provider entity's vendor rating.
   
c) The discovery service returns one or more Web service descriptions (or references to them) that meet the specified criteria. If multiple service descriptions are returned, the requester entity selects one, perhaps using additional criteria.
2. The requester and provider entities agree on the semantics ("Sem") of the desired interaction. Although this may commonly be achieved by the provider entity defining the semantics and offering them on a take-it-or-leave-it basis to the requester entity, it could be achieved in other ways. For example, both parties may adopt certain standard service semantics that are defined by some industry standards body. Or in some circumstances the requester could define the semantics. The important point is that the parties must agree on the semantics, regardless of how that is achieved.

This step also requires that the parties agree on the service description that is to be used. However, since the requester entity obtained the Web service description in Step 1.c, in effect the requester and provider entities have already done so.
WS Discovery [3]

3. The service description and semantics are input to, or embodied in, both the requester agent and the provider agent, as appropriate.

4. The requester agent and provider agent exchange SOAP messages on behalf of their owners.
Manual vs. Autonomous WSD [3]

- The discovery process described above is not specific about who or what within the requester entity actually performs the discovery. Under *manual discovery*, a requester *human* uses a discovery service (typically at design time) to locate and select a service description that meets the desired functional and other criteria. Under *autonomous discovery*, the requester *agent* performs this task, either at design time or run time. Although the steps are similar in either case, the constraints and needs are significantly different, such as:
  - *Interface requirements*. The requirements for something that is intended for human interaction are very different from the requirements for something that is intended for machine interaction.
  - *Need for standardization*. There is far less need to standardize an interface or protocol that humans use than those that machines are intended to use.
  - *Trust*. People do not necessarily trust machines to make decisions that may have significant consequences.

- In the case of autonomous discovery, the need for machine-processable semantics is greatly increased.
Delivering of SOA with WS

• Basic building blocks
  – XML
  – SOAP
  – WSDL
  – UDDI

• Advanced aspects
  – Security
  – Composition
  – Transactions etc.
Building Blocks [1]

Interpretation of the core specifications and links through the WS-I Basic Profile 1.1
XML in Brief [1]

XML [XML]
- Markup language composed of tags and data
- Elements and attributes
- Read by an XML processor
- Requires grammar definition
- Valid and well-formed

XML Namespaces [XMLNS]
- Qualified names: prefix and local parts
- Global naming mechanism for XML
- Multiple namespaces in same document

XML Schema [XMLSch]
- Provides grammar for XML instance docs
- Built-in types
- Simple and complex custom data types

XML instance document example

XML document grammar and valid XML instances
SOAP introduction [1]

- SOAP message elements: Envelope, Headers, Message Body and Faults
- Two communication styles: Document style, RPC style
- Literal or SOAP encoding of message body plus attachments support
WSDL Structure [2]

Where to access it

How to access it

Service

Port (e.g. http://host/svc)

Binding (e.g. SOAP, JMS, direct call)

Port

Binding

Abstract interface

portType

operation(s)

inMessage

outMessage

Abstract definition of the service (set of Operations)

Operation: An Exchange of Messages between a service Requestor and a Service provider
References

  http://www.perspectivesonwebservices.de/download/t48-tomlinson-zimmermann-finalExt.ppt


  http://www.w3.org/2005/Talks/1115-hh-k-ecows/