SOA Architecture and Concepts

李银胜
liys@fudan.edu.cn, eb.fudan.edu.cn

SOFT130048.01 周一 2-4 节 Z2102
• LECTURE1. About the Course
• LECTURE2. The Case & the Project
• LECTURE3. SOA Architecture & Concepts (*)
• LECTURE4. Technical Basis: XML with SOA (*)
• LECTURE5. Basic Techniques: Web Services (*)
• LECTURE6. Service-Oriented Application Techniques (*)
• LECTURE7. Service-Oriented Implementation (*)
~ Android Development Tutoring on Demand (>=3 times)
* Class Presentation of Prj advances
@ Challenges: How to

- structure and use the services?
- understand SOA?
- using Web services in an app?
- design an app and identify services?
- compose and flow the services?

@ Assign …
Introduction: Evolvement of W3C Web Service Architecture

Service-Oriented Architecture and Concepts >> Architecture

Figures 1

1. Agree on semantics & WSD
2. Input Semantics & WSD
3. Interact
4. Interact
Service-Oriented Architecture and Concepts

Architecture: W3C by Feb 11, 2004

20040211
@ Who use Web services

Machine/Software or humans?
@ Using a Web service

There are many ways that a requester entity might engage and use a Web service. In general, the following broad steps are required, as illustrated

(1) the requester and provider entities become known to each other (or at least one becomes know 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; and
(4) the requester and provider agents exchange messages, thus performing some task on behalf of the requester and provider entities. (I.e., the exchange of messages with the provider agent represents the concrete manifestation of interacting with the provider entity's Web service.)
Registry Agency Required?

To review the SOA evolving from triangle into two entities?
Architecture: With Discovery Service
Architecture: One for human discovery, by ...
Architecture: One for autonomous Selection, by ...
Architecture: Goals and Non-goal by W3C

To Promote:

- interoperability between Web services,
- integration with the World Wide Web,
- reliability of Web services,
- security of Web services,
- scalability and extensibility of Web services,
- manageability of Web services.
Web Service

&

Web

?
Service-Oriented Architecture and Concepts

Architecture: Goals and Non-goal by W3C

Not:

✓ to prescribe a specific programming model or programming technology
✓ to constrain the internal architecture and implementation of specific Web services
✓ to demonstrate how Web services are constructed
✓ to be specific about how messages or other descriptions are formatted
✓ to determine specific technologies for messaging, discovery, choreography etc.

@ For What?
@ Challenges: How to

• structure and use the services?
• understand SOA?
• using Web services in an app?
• design an app and identify services?
• compose and flow the services?

@ Assign …
A service is a set of actions that form a coherent whole from the point of view of service providers and service requesters.
Service-Oriented Architecture and Concepts

@ Service definition by W3C

Relationships to other elements:

- A service performs one or more tasks
- A service has a service description
- A service has one or more service providers
- A service has zero or more service requesters
- A service has an identifier
- A service has a service semantics
- A service has a service interface
- A service is realized by one or more agents acting as service providers
- A service is invoked by exchanging messages
- A service has a service execution model
Web Service

✓ A software module. It is deployed on the provider platform to be accessed via network

✓ It can also be the requestor when it need other WS to implement its function
A Web service as “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”
Opinions for Web Services (1)

Gartner (2001) for instances, defines Web services generically as “loosely coupled software components delivered over the Internet via standards-based technologies.”
Opinions for Web Services (2)

Microsoft (2003)

emphases more on Web services’ XML, Web and application-oriented features in its latest documentations, “XML Web services extend the World Wide Web infrastructure to provide the means for software to connect to other software applications. Applications access Web Services via ubiquitous Web protocols and data formats such as HTTP, XML, and SOAP, with no need to worry about how each Web service is implemented. Web services combine the best aspects of component-based development and the Web.”

Ms .NET is a complete technical framework for Web services development application
Opinions for Web Services (3)

IBM (2003) explains Web services more comprehensively and touches on their orchestration and service-oriented essence: “Web services is a technology that allows applications to communicate with each other in a platform- and programming language-independent manner. A Web service is a software interface that describes a collection of operations that can be accessed over the network through standardized XML messaging. It uses protocols based on the XML language to describe an operation to execute or data to exchange with another Web service. A group of Web services interacting together in this manner defines a particular Web service application in a Service-Oriented Architecture.”

IBM will update all its products to support Web services technologies. Its software family will be migrated up to Web services era. Currently IBM has five Web services software products: Websphere, Visual Age for Java, DB2, Tivoli, and Lotus.
Opinions for Web Services (4)

Sun (2003)

A modular, well-defined, encapsulated function
- Used for loosely coupled integration between applications or systems
- Based on XML, transported in two forms:
  - Synchronous (RPC): SOAP
  - Asynchronous (messaging): SOAP and ebXML Messaging
- Sometimes advertised and discovered in a service registry
  - UDDI, ebXML Reg/Rep, and WSDL are popular standards
- Over Intranet and Internet

SunONE:
- Sun Open Net Environment (Sun ONE) is Sun's standards-based software vision, architecture, platform, and expertise for building and deploying Services on Demand. It provides a highly scalable and robust foundation for traditional software applications as well as current Web-based applications, while laying the foundation for the next-generation distributed computing models such as Web services.
- Simply put, SunONE is J2EE with Web Services supports
- The objective is to make network accessible for any information device at any time
Architecture: Roles@Agents

- Agents

A Web service is viewed as an abstract notion that must be implemented by a concrete agent. The agent is the concrete entity (a piece of software) that sends and receives messages, while the service is the abstract set of functionality that is provided. To illustrate this distinction, you might implement a particular Web service using one agent one day (perhaps written in one programming language), and a different agent the next day (perhaps written in a different programming language). Although the agent may have changed, the Web service remains the same.
Service-Oriented Architecture and Concepts

Architecture: Roles

Service Provider: the owner of the service from the viewpoint of business. The platform to be entrusted for access from the viewpoint of architecture. The purpose of a Web service is to provide some functionality on behalf of its owner -- a person or organization, such as a business or an individual. The provider entity is the person or organization that provides an appropriate agent to implement a particular service (W3C).

Service Requestor: Specified function requestor from the viewpoint of business. The application to look up, invoke and interact with WS from the viewpoint of architecture. A requester entity is a person or organization that wishes to make use of a provider entity's Web service. It will use a requester agent to exchange messages with the provider entity's provider agent. In order for this message exchange to be successful, the requester entity and the provider entity must first agree on both the semantics and the mechanics of the message exchange (W3C).

Service Registry: Optional role of WS architecture for requestors who require static binding
Architecture: Roles@Web Service Description

- WS Description (WSD)
  ✓ Includes service interface and implementation details, e.g., data types, operations, binding and URI of a WS
  ✓ Can be published to requestors or discovery agency
  ✓ The mechanics of the message exchange are documented in a Web service description (WSD). The WSD is a machine-processable specification of the Web service's interface. It defines the message formats, datatypes, transport protocols, and transport serialization formats that should be used between the requester agent and the provider agent. It also specifies one or more network locations ("endpoints") at which a provider agent can be invoked, and may provide some information about the message exchange pattern that is expected (W3C).
Architecture: Roles@Semantics (1/2)

- Semantics

✓ The semantics of the message exchange represents the "contract" between the requester entity and the provider entity regarding the purpose and consequences of the interaction. It also includes any additional details on the mechanics of the message exchange that are not specified in the service description. Although this contract represents the overall agreement between the requester entity and the provider entity on how and why their respective agents will interact, it is not necessarily written or explicitly negotiated. It may be explicit or implicit, oral or written, machine processable or human oriented.

✓ While the service description represents a contract governing the mechanics of interacting with a particular service, the semantics represents a contract governing the meaning and purpose of that interaction.
Service-Oriented Architecture and Concepts >> Concepts

Architecture: Roles@Semantics (1/2)

- A Semantics is

✓ the contract between the service provider and the service requester concerning the effects and requirements pertaining to the use of a service
✓ about the service tasks that constitute the service.
✓ (may be) expressed in a service description language
✓ (may be) identified in a service description
✓ describes the intended effects of using a service
✓ describes the relationship between the service provider and the service requester
Architecture: Roles@Humans

- **Humans**: although one of the main purposes of Web services is to automate processes that might otherwise be performed manually, humans still play a role in their architecture and use, notably in two ways:

  ✓ Humans need to agree on the semantics and the service description.

  ✓ Humans create the requester and provider agents (either directly or indirectly).
Architecture: Operations

- Publish
  ✓ Direct publish. Providers send WSDL description directly to requestors
  ✓ Registration publish. Providers publish WSDL description to local WSDL registry, private UDDI registry or UDDI agency

- Discovery: Registry, Index or Peer-to-Peer
  ✓ Lookup WS descriptions to develop in design
  ✓ Lookup WS descriptions for their binding and location to invoke in execution

- Bind, Invoke or Interact
  ✓ Requestors use bindings in WS descriptions to locate, contact and invoke them. Invoke and interact with WS when executing
Architecture: Scenario

✓ WS providers define WS descriptions and publish them to service provider or service registry

✓ WS requesters lookup their intended WS descriptions at local or registration agencies

✓ Requesters analyze the found WS descriptions, and bind and interact with the provided services
Architecture: Lifecycle

- Construction
  ✓ Develop and debug WS, define WS interface description and implementation details

- Deployment
  ✓ Publish WS interfaces and implementation definition to requestors or discovery agencies. Deploy and configure its executable to execution environments

- Invocation and Execution
  ✓ Having been deployed, functional, and accessible via network. Requestors can now discover and bind with it

- Management
  ✓ Manage and maintain WS applications. The concerns for this phase include security, flexibility, performance, QoS and business process
Features of SOA

- Granularity
- Loosely-coupled
- Open standard interface
Service Composition and Integration

✓ A distributed system can be viewed as collaborative integration environment with services.
✓ Services are not treated as isolated and one-time affairs but rather as elements in an interactive and dynamic collaboration structure.
✓ Service collaborations within or across environments are modeled in terms of supported transactions and processes. These collaborations are subject to norms and protocols specified for business domains. Services are thereby orchestrated vertically within one, or horizontally across multiple environments.
@ Challenges: How to

• structure and use the services?
• understand SOA?
• using Web services in an app?
• design an app and identify services?
• compose and flow the services?

@ Assign …
Description:

– A portal application running within an application server aggregates information from multiple internal applications, providing a single point of entry into business processes spread across those applications.

– The portal application gets information about Web Services offered by internal applications using private UDDI registry and invokes these services over the intranet.

– Binding information for frequently used Web Services can be cached by the application, to avoid the resource intensive and time consuming dynamic binding.

– In this case, the Web Services loosely integrate portal with CRM and ERP applications.
Service-Oriented Architecture and Concepts

Scenario:

Step 1: After logging on to the company portal, users request information.
Scenario:

Step 2: The application supporting the portal framework gets information about Web Services made available by the CRM and ERP applications by doing a look up in the private UDDI registry.
Step 3 : The location and WSDL binding info of Web Services is sent to the application server.
Step 4: The application invokes the Web Service published by the CRM application and retrieves the personal information, such as name, social security number, mailing address and email, of the user. The communication is based on SOAP.
Step 5: The application invokes the Web Service published by the ERP application and retrieves the account information, such as account number, balance and transaction history, of the user. The communication is based on SOAP.
Scenario:

Step 6: The information is then formatted and sent to the user.
@ Challenges: How to

• structure and use the services?
• understand SOA?
• using Web services in an app?
• design an app and identify services?
• compose and flow the services?

@ Assign …
Case of WebGIS

To illustrate how to design an app and identify services?
Service-Oriented Architecture and Concepts

WS-based Web GIS Applications

Applications

Call WS to operate on data

GI---WS1

Search Metadata and acquire

Registry

Service Metadata Update

GI---WS2

Direct Retrieval

Data Access and store

Other Data
Coverage Data
Feature Data

WS-based Web GIS Applications
• ISO TC211 → ISO19119 is a abstract specification for graphic information Web services

• ISO19119 defines concepts and architecture of GI Web services
A service is a set of operations that can be accessed through an interface. Users can get responses by trigger its activity.

Typical GI services are: get data, display data, identification and coordinates translation, etc.
ISO19119 classifies services into six categories:

<table>
<thead>
<tr>
<th>Service category</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Human Interaction</td>
<td>Services for managing user interfaces, graphics, multimedia, and presenting compound documents</td>
</tr>
<tr>
<td>Information Management</td>
<td>Services for managing the development, manipulation, and storage of metadata, conceptual schemas, and datasets</td>
</tr>
<tr>
<td>Workflow</td>
<td>Services that support specific tasks or work-related activities.</td>
</tr>
<tr>
<td>Processing</td>
<td>Services that perform large-scale computations; a processing service does not include capabilities for providing persistent storage of data or transfer of data over networks</td>
</tr>
<tr>
<td>Communication</td>
<td>Services that encode and transfer data across networks</td>
</tr>
<tr>
<td>System Management</td>
<td>Services for managing system components, applications, and networks (including access control)</td>
</tr>
</tbody>
</table>
## OGC service classification scheme

<table>
<thead>
<tr>
<th>OGC code</th>
<th>Service class</th>
<th>OGC code</th>
<th>Service class</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>OGC web service [ROOT]</td>
<td>2600</td>
<td>Sensor access</td>
</tr>
<tr>
<td>1000</td>
<td>Human interaction</td>
<td>2700</td>
<td>Order handling</td>
</tr>
<tr>
<td>1100</td>
<td>Portrayal</td>
<td>3000</td>
<td>Workflow</td>
</tr>
<tr>
<td>1110</td>
<td>Geospatial viewer</td>
<td>3100</td>
<td>Chain definition</td>
</tr>
<tr>
<td>1111</td>
<td>Animation</td>
<td>3200</td>
<td>Enactment</td>
</tr>
<tr>
<td>1112</td>
<td>Mosaicing</td>
<td>3300</td>
<td>Subscription</td>
</tr>
<tr>
<td>1113</td>
<td>Perspective</td>
<td>4000</td>
<td>Processing</td>
</tr>
<tr>
<td>1114</td>
<td>Imagery</td>
<td>4100</td>
<td>Spatial</td>
</tr>
<tr>
<td>1120</td>
<td>Geospatial symbol editor</td>
<td>4110</td>
<td>Coordinate conversion</td>
</tr>
<tr>
<td>1130</td>
<td>Feature generalization editor</td>
<td>4120</td>
<td>Coordinate transformation</td>
</tr>
<tr>
<td>1200</td>
<td>Service interaction editor</td>
<td>4130</td>
<td>Representation conversion</td>
</tr>
<tr>
<td>1300</td>
<td>Registry browser</td>
<td>4140</td>
<td>Orthorectification</td>
</tr>
<tr>
<td>2000</td>
<td>Information Management</td>
<td>4150</td>
<td>Subsetting</td>
</tr>
<tr>
<td>2100</td>
<td>Feature access</td>
<td>4160</td>
<td>Sampling</td>
</tr>
<tr>
<td>2200</td>
<td>Coverage access</td>
<td>4170</td>
<td>Feature manipulation</td>
</tr>
<tr>
<td>2210</td>
<td>Real-time sensor</td>
<td>4180</td>
<td>Feature generalization</td>
</tr>
<tr>
<td>2300</td>
<td>Map access</td>
<td>4190</td>
<td>Route determination</td>
</tr>
<tr>
<td>2400</td>
<td>Gazetteer</td>
<td>41A0</td>
<td>Positioning</td>
</tr>
<tr>
<td>2500</td>
<td>Registry</td>
<td>4200</td>
<td>Thematic</td>
</tr>
<tr>
<td>OGC code</td>
<td>OGC code</td>
<td>Service-Oriented Architecture and Concepts</td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td>----------</td>
<td>-------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>4210</td>
<td>4300</td>
<td>Geoparameter calculation</td>
<td></td>
</tr>
<tr>
<td>4220</td>
<td>4310</td>
<td>Temporal</td>
<td></td>
</tr>
<tr>
<td>4221</td>
<td>4320</td>
<td>Thematic classification</td>
<td></td>
</tr>
<tr>
<td>4222</td>
<td>4330</td>
<td>Unsupervised</td>
<td></td>
</tr>
<tr>
<td>4230</td>
<td>4340</td>
<td>Supervised</td>
<td></td>
</tr>
<tr>
<td>4240</td>
<td>4400</td>
<td>Change detection</td>
<td></td>
</tr>
<tr>
<td>4250</td>
<td>4410</td>
<td>Radiometric correction</td>
<td></td>
</tr>
<tr>
<td>4260</td>
<td>4420</td>
<td>Geospatial analysis</td>
<td></td>
</tr>
<tr>
<td>4261</td>
<td>5000</td>
<td>Image processing</td>
<td></td>
</tr>
<tr>
<td>4262</td>
<td>5100</td>
<td>Reduced resolution generation</td>
<td></td>
</tr>
<tr>
<td>4263</td>
<td>5200</td>
<td>Image manipulation</td>
<td></td>
</tr>
<tr>
<td>4270</td>
<td>5300</td>
<td>Image synthesis</td>
<td></td>
</tr>
<tr>
<td>4280</td>
<td>6000</td>
<td>Geoparsing</td>
<td></td>
</tr>
</tbody>
</table>

> Example for design
Also ISO19119 defines service links and service metadata.

OGC takes ISO19119 as foundation of abstract specification of service model.
Each service type is identified by the set of operations and the signature (e.g., inputs, outputs, exceptions) of each operation comprised the interface.

From an OO programming perspective, services do specialized one another through extension (or restriction) of their interfaces and the operations of the interfaces and the signature (e.g., inputs, outputs, exceptions) of individual operations.

Services organized according to an interface inheritance hierarchy can provide (at least part of) the basis for deciding if a service of one type may be composed with another to conduct some interaction.

Next Figure illustrates the basic service model for inheritance hierarchies of service by interfaces and operations.
Service descriptions admit several levels of type management: a service type comprises one or more interface types plus any number of service properties, an interface type encapsulates one or more operation types, and so on.
Service registry interfaces typically enable the following basic set of capabilities:

- Add, delete, modify, and query service offers
- Add, delete, and query type descriptions
- Classify registry objects in a flexible manner

In addition to the above minimal set of capabilities, there are a number of IPSM-specific requirements that encompasses the following registry interactions:

- Submitting a new classification scheme
- Extending an existing classification scheme
- Modifying the classification of a registered service offer
- Updating the binding specification (i.e., WSDL implementation document) for a published service offer
- Modifying the data description associated with a service offer
OGC classifies the implementation specifications of GI Web services into 4 categories based on provided services.
C1: Core Service. Common interfaces regardless of application fields. They support services of other applications.

These specifications include:
- Coordinate translation
- Directory
- Service registry…
C2: Web Mapping Service. These specifications allow Web space info to dynamic query, access, translation and aggregation between servers developed by the vendors that conform to OpenGIS. OpenGIS specs include Web Map Server specification (accepted), Web Feature Server specification (accepted), etc.
C3: Location Service. These services depend on gateway services, which integrate location app services with common mobile terminal, wireless platform, IP platform, and mobile location identification system (eg. GPS)
OGC's open architecture for location services will be based on interfaces and protocols that support “Core Services” including:

• Gateway Services that integrate OpenLS location application services with position determination equipment in the MPC/GMLC (the place in the network that manages the location of devices).

• Directory services for searching yellow pages, green pages, travel guides, and so on.

• Route determination services for navigation.

• Geocode (address to X,Y) and reverse geocode (X,Y to address) services.

• Map/feature display services.
C4: Geospatial Fusion Services. They fuse address, space name, coordinates, image spots, descriptive direction into a information management framework, which support find, discovery and share non-map formatted space info. They include Gazetteer Service Interface and Geocoder Service, etc.
The above application model has 4 components:

User Applications. Software that users interact. They are usually customizable analysis software or domain applications, or generic navigators. They get input directly from data repository, or from middleware services. Middleware services preprocess the data.

GI Web Services. Core services of the interoperation stack. Probably they draw map based on raw data, or conduct high-level analysis e.g. abstracting features or coordinate translation. They provide data, maps or other inputs to user applications or other services (this style of service is so-called “service chain”).
Data repository. It provides other applications or services with geographic space data of Feature, Coverage and data objects or tables. Please refer to Web Map Server specification for details.

Directory center. Allow customers and services to find available data or services. Service and data servers must be registered with directory server. Registry will be updated when services or data get changed. This directory can use UDDI to register common GI Web services, or use proprietary directory to store application-specific Web services.
The model is executed as follow

1) Data storage register itself or update its metadata

2) Develop Web services for data operation and processing. They are described in WSDL and registered or updated at directory center.

3) Applications search data through directory center. The found data could be retrieved, or

4) Applications can search a Web service to handle the data.

5) Applications invoke the Web Service to operate or process the data.
This mode is well-performed in cross-platform since Web services are used to compose functions and SOAP is used as the communication protocol among components.

Moreover, it facilitates system development, debug and management since component-based development is applicable. The system is therefore flexible or reusable.
@ Challenges: How to

• structure and use the services?
• understand SOA?
• using Web services in an app?
• design an app and identify services?
• compose and flow the services?

@ Assign …
Features of SOA

• Well-defined granularity
• Self-contained
• Reuse and extensible
• Open standard interface
• Many-Many, R/P
• Loosely-Coupled
A e-Business case: Travel reservation (1/23)

A typical scenario involving a set of Web services to serve as an useful example for Web services and to generate requirements and demonstrate its applications

Description

✓ A company (travel agent) wants to offer to people the ability to book complete vacation packages: plane/train/bus tickets, hotels, car rental, excursions, etc.
✓ Service providers (airlines, bus companies, hotel chains, etc) are providing Web services to query their offerings and perform reservations.
✓ Credit card companies are also providing services to guarantee payments made by consumers.
✓ Due to the loosely coupled-nature of Web services, the travel agent doesn’t need to have a priori agreements with service providers or credit card companies. This allows the travel agent to have access to more services, offering more options to its customers, the credit card companies to offer their services broadly and therefore make their customers happy, and the service providers can offer their services broadly and easily and therefore generating more business for themselves.
Scope

✓ For this version of the usage scenario, we will limit ourselves to booking of vacation packages.
✓ We will assume that cancellation is not possible once a package has been purchased.
A e-Business case: Travel reservation (2/23)

**Stakeholders / Interests**

✓ Travel agent: provides a system to provide the user with options for his/her vacation; earns money by charging fees for each package bought.

✓ Service providers: sell their services by making them available widely.

✓ Credit card company: enable customers to use their credit cards in a very large number of cases; make profit with each money transaction.

✓ Consumer: book vacation easily by choosing among a large variety of offers.

✓ Only the user in the scenario is a human being. The travel agent service, airline, hotel and payment services that the travel agent service is interacting with, are machines.
A e-Business case: Travel reservation (3/23)

Actors & Goals

✓ Consumer: best combination of services and prices for his/her needs.
✓ Travel agent: customer satisfaction, sell packages.
✓ Service providers: sell services.
✓ Credit company: qualify buyer, do the payment.

Use Cases

✓ The following use cases describe how a user would make a reservation for a vacation package (flight and hotel room).
A e-Business case: Travel reservation (4/23)

Overview.

Customer uses Travel Agent Service

- Credit Card WS 1
- Airline WS 1
- Airline WS 2
- Airline WS 3
- Hotel WS 1
- Hotel WS 2
- Hotel WS 3
A e-Business case: Travel reservation (5/23)

Composing a vacation package

- My hotel
- Credit Card Company 3
- Credit Card Company 2
- Credit Card Company 1

Customer → Travel Agent Service

Travel Agent Service →
- Airline 1
- Airline 2
- Airline 3
- Hotel 1
- Hotel 2
- Hotel 3

Ontology

queries → Airline 1, Airline 2, Airline 3, Hotel 1, Hotel 2, Hotel 3
Service-Oriented Architecture and Concepts >> Example for flows

Booking a chosen vacation

- My hotel
  - Hotel 3
  - Hotel 1
- Credit Card Company 3
- Credit Card Company 1
- Airline 2
- Airline 1
- Customer

uses

Travel Agent Service

interacts with

books

interacts with

gets payment

interacts with

pays with

assembles

receives

gets payment

gets payment

Vacation package
Introducing a new hotel Web

Service-Oriented Architecture and Concepts

Example for flows

Credit Card Company 3
Credit Card Company 2
Credit Card Company 1

Customer uses Travel Agent Service

queries queries queries queries queries

Airline 1 Airline 2 Airline 3 Hotel 1 Hotel 2 Hotel 3

Ontology

reads describes

Description

My hotel
Service-Oriented Architecture and Concepts

Booking the new cheaper option

- Hotel 1
- Hotel 2
- Hotel 3
- Credit Card Company 1
- Credit Card Company 3
- Airline 1
- Airline 2

Customer uses Travel Agent Service

- interacts with Airline 3
- interacts with My hotel
- reads Description
- receives Vacation package
- assembles

Customer pays with
- gets payment
- interacts with Airline 3
- interacts with My hotel
- gets payment

Example for flows
An assumption for this usage scenario is that all the services are using common concepts (e.g. flight, economy class, room, etc).

For the travel agent service to understand the airline services and to be able to send meaningful information to them, a travel industry ontology needs to exist and be used by the Web services taking part in this scenario.

An ontology is a formal description of a set of concepts and their relationships to each other. By associating a name with each concept, an ontology defines a standard vocabulary that can be used to communicate those concepts.
A e-Business case: Travel reservation (10/23)

**Noted:** Some additional technology is needed for this usage scenario:

- ✓ Context maintenance.
- ✓ Reliability: In order to make money, each step needs to happen.
- ✓ Trust mechanisms for the services to do business with each other.
- ✓ Description of orchestration of services: If a reservation of a flight involves interacting with a couple of Web services, the airline would document in a machine readable way how to interact with the two single services in order to get the desired result, including how to handle errors in the process fails before the operation is completed.
- ✓ ... ...
A e-Business case: Travel reservation (11/23)

Note that this usage scenario could be different in the following ways:

- the user could have bought some travel agent software; the travel agent service could reside locally on his/her computer.
- the user could write tools to interact directly with the airline and hotel services.
A e-Business case: Travel reservation(12/23)

1. User requests availabilities about some travel dates

Goal / Context

The user gets the location of a travel agent service via an unspecified way (search engine, URI in an email, service directory, etc).

The user provides a destination and some dates to the travel agent service. The travel agent service inquires airlines about deals and presents them to the user.
A e-Business case: Travel reservation(13/23)

Scenario / Steps

• The user is presented with a form to fill in order to provide the travel agent service with details about dates of his/her travel and the destination.
• The user submits the information to the service in order to get a list of flights corresponding to his/her schedule.
• The travel agent service finds a list of airlines.
• For each airline found:
  ✓ The travel agent service requests a description of how to communicate with the service found.
  ✓ The travel agent service requests a list of flights accommodating the user.
• The travel agent service presents the results of the queries to the user letting him choose the best option.
Extensions
If no flight can be found, the user should be presented with an error.

Technologies / Requirements
Discovery technology: used by the travel agent service to find the airlines services.
Description language: used by the airlines to describe their query services to the travel agent service.
Response to queries: XML documents that the travel agent service processes and merge together.
Ontologies: the data coming from different airline services and expressed with different XML vocabularies needs some semantics to be merged in a meaningful way.
2. User requests chooses flight and looks for hotels

Goal / Context

The user has been presented with options for flights to go to his/her destination.

The user chooses a preferred flight.

The service puts the seats on hold, and goes on with proposing lodging options to the user.
A e-Business case: Travel reservation (16/23)

Scenario / Steps

- The user communicates his/her choice for the flight.
- The travel agent service requests the chosen airline to put the flight on hold:
  - The travel agent service requests a description of how to put a seat on hold to the airline service.
- The travel agent service sends the request accordingly.
- The airline returns a confirmation number with an expiry date.
- The travel agent service finds a list of airlines.
- For each hotel found:
  - The travel agent service requests a description of how to communicate with the service found.
  - The travel agent service requests accommodation options for the period.
- The travel agent service looks for payment services available, and builds a list of options for the user.
- The travel agent service presents the results of the queries to the user letting him choose the best option, along with the payment options offered.
A e-Business case: Travel reservation(17/23)

Extensions
If the seats chosen are not available anymore, the travel agent service presents the user with an error message and the user is presented with an updated list of available flights to choose from.

Technologies / Requirements
Description language: used by the airlines to describe their services to put tickets on hold to the travel agent service, by the hotels to describe their query services to the travel agent service.

Discovery technology: used by the travel agent service to find the hotels services.

Ontologies: the data coming from different accommodation services and expressed with different XML vocabularies needs some semantics to be merged in a meaningful way.
A e-Business case: Travel reservation

3. User books hotel room and flight

Goal / Context

The user has been presented with options for hotels to go to his/her destination and a means of payment.

The user chooses a hotel option.

The travel agent service contacts a bank for payment authorization. The service books the hotel and confirms the flight, using the payment authorization from the bank.
A e-Business case: Travel reservation (19/23)

Scenario / Steps

• The user communicates his/her accommodation choice to the travel agent service.
• The travel agent service contacts the bank service that the user chose to confirm payment:
  ✓ The travel agent service requests a description of how to guarantee payment of the total amount.
  ✓ The travel agent service sends the request accordingly.
  ✓ The response indicates success with an authorization number, signed by the payment authority.
• The travel agent service books the hotel room:
  ✓ The travel agent service requests a description of how to book a room to the chosen hotel service.
  ✓ The travel agent service sends a request in order to find out how to cancel the reservation should a problem occur later in the process.
  ✓ The travel agent service sends the request accordingly, communicating the payment service chosen and the signed authorization number from this service.
A e-Business case: Travel reservation (20/23)

- The travel agent service confirms the flight reservation:
  - The travel agent service requests a description of how to buy a ticket on hold to the airline service.
  - The travel agent service sends a request in order to find out how to cancel the reservation should a problem occur later in the process.
  - The travel agent service sends the request accordingly, communicating the payment service chosen and the signed authorization number from this service.
- The travel agent service charges a fee to the user:
  - The travel agent service requests a description of how to request payment to the payment service.
  - The travel agent service sends the request accordingly, along with the authorization number signed by the payment service.
- The service provides the user with various confirmation numbers and wishes the user a good vacation.
A e-Business case: Travel reservation (21/23)

Extensions

- If the payment service doesn't confirm the validity of the user's payment option, the user should be presented with an error.
- If the hotel room cannot be booked, the user should be presented with an error and should get to choose from an updated list of options.
- If the flight reservation cannot be confirmed, the hotel room reservation should be canceled and the user should be presented with an error and start the reservation process again.
Technologies / Requirements

Service description technology: used by the payment authority to describe its confirmation service, by the hotel to describe its room booking service, and by the airline to describe its service to buy tickets by confirming seats on hold.

Authentication technology: used by the payment authority to sign the payment authorization to be trusted by the hotel service, the airline service and the travel agent service.

Encryption technology: used by the payment service and the travel agent service to communicate the user’s payment information confidentially.

Ontologies: the payment confirmation needs to be used in a way meaningful to the travel service, hotel and airline services; in other words, the output of one service needs to be used as the input to other services that might use different vocabularies.
A e-Business case: Travel reservation

Notes on the scenario

This scenario illustrates how a program, the travel agent service, can interact dynamically with airline services, hotel services, without a priori knowledge of them or of the way they work. Thanks to the ontologies used, the program can adapt to variations of formats that an airline service might be using and adapt to the introduction of new products.

However, there is a limit to what the travel agent service can understand. For example, it is likely to be able to understand the introduction of a new class of tickets, say class Z. However, if the restrictions on class Z tickets use concepts that it is not aware of (say that class Z tickets can only be bought more than 60 days in advance and with a valid international student identification), the developers of the travel agent service will need to implement the extra logic to make it understand this new type of restriction, including validating the student identification.
A e-Business case: Travel reservation

Review the travel service:

1. Decompose into elementary services
2. Describe elementary services by goals instead of hardwiring them.
3. Keep the human programmer out of the loop to keep it economic, on demand, and scalable.
4. You cannot achieve this vision without semantic web technology that maintains selection and combination of heterogeneous web services during runtime.
@ Challenges: How to

• structure and use the services?

• understand SOA?

• using Web services in an app?

• design an app and identify services?

• compose and flow the services?

@ Assign …
S1. To select an interesting topic
S2. To define the APP features and functions

S3. (To get a template and ) identify the Web services
S4. (To run the template and ) design the architecture
S5. To design the critical flows of the app
S6. To describe a Web service with XML
S7. To define the schema of the Web service
S8. To develop the basic function of the app
S9. To develop the search function of the app
S10. To develop the package function of the app
• What is the relationship between Web services and SOA?
• Explain why SOA/Web Services can lead the distributed computing techniques?
• What are the four steps to use a Web service in the SOA specification?
• What are the operations in SOA?
• How to become known to each other for service provider and requestor?
• Is service registry required or optional?
• For a human, how to find a service?
• How to find a service autonomously?
• Try to illustrate several goals and their purposes and consequences for SOA to promote.
• Try to illustrate several goals and their purposes and consequences for SOA not to promote.
• Explain the relationship between Web services and Web.
• What is a service by W3C?
• What is a Web service by W3C?
• Please provide at least 5 relationships between a service with other distributed concepts.
• Who use Web services, human or machine?
• What is used to communicate between WS?
• What is used to describe a Web service?
• What is used to implement a Web service?
• What is used to model Web services processes?
• What is used to register a Web service
• Try to illustrate 3 of the key SOA features comparing traditional distributed concepts.
• According to W3C, is this correct that a service has a service semantics?
• What is a Web service by W3C?
• What is the agents in SOA? Are they are the ones in MAS?
• What viewpoint should be based on to under the Service Provider and the Service Requestor?
• What is the Service Provider from the viewpoint of business?
• What is the Service Requestor from the viewpoint of business?
• What details can Web Service description carry?
• What is a Semantics and the details it includes?
• Do humans have a role in Web services architecture? If yes, what is the role?
• Try to explain the SOA's lifecycle.
• Where can Web services used? Please illustrate 2 scenarios.
Take a break ...