Web Services and Business Processes
Infrastructure for agile SW deployment

M.I. Capel

ETS Ingenierías Informática y
Telecomunicación
Departamento de Lenguajes y Sistemas Informáticos
Universidad de Granada
Email: manuelcapel@ugr.es
http://lsi.ugr.es/mcapel/

November, 15th 2023
Máster Universitario en Ingeniería Informática

Introduction to Middleware
Introduction to SOA
Business process modeling
Web services composition

Middleware characteristics

Definition of Middleware

software that offers the capability of integration and reuse of components at different scale (methods, protocols, software and systems) on demand, aimed to abstract the distribution and heterogeneity of the available services to the underlying computing environment
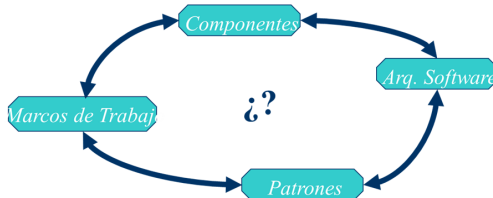

Figure: Middleware is something that's referred to by software developers as "software glue"

Introduction to Middleware
Introduction to SOA
Business process modeling
Web services composition

Middleware characteristics-II

Middleware objectives

To support the development, deployment, execution and maintenance of software applications, by saving the distance between the application layer itself and the underlying layers (operating system, network stack and hardware)
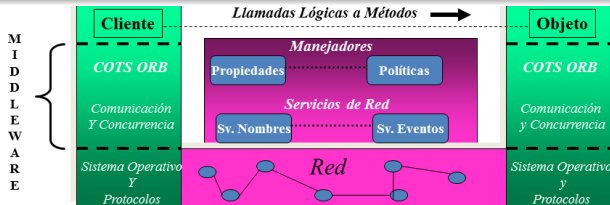

Figure: Resides between the application, operating systems, protocols and underlying hardware

Introduction to Middleware
Introduction to SOA
Business process modeling
Web services composition

Middleware characteristics-III

Middleware example

The functionality related to gesture or voice recognition is normally processed by a specific intermediary software and the results of that processing are transmitted to user applications

Introduction to Middleware
Introduction to SOA
Business process modeling
Web services composition

Support technologies for component-based software and Systems development

Modelos basados en objetos distribuidos	Sistemas basados en Infraestructuras componentes software (Middleware)	Sistemas Basados en Agentes	Sistemas Cliente-Servidor	Transportabilidad
				Extensibilidad
				Concurrencia
				Confiabilidad
				Apertura
				Interoperabilidad
				Reconfiguración Dinámica
				Autoadaptación
				Colaboración

Figure: Covering degree of the required properties

Introduction to Middleware
Introduction to SOA
Business process modeling
Web services composition

Middleware shortcomings

Disavantages

- Loss of speed
    - Slow computer for application server
    - slow connection between applications server and database server
- Security
    - Can introduce new security holes
- Reliability
    - Have to worry about an extra computer going down

Introduction to Middleware
Introduction to SOA
Business process modeling
Web services composition

Middleware shortcomings-II

High requirements, difficult to fulfill by applications with time-criticality

- Any middleware-based solution must provide a runtime environment that is responsible for coordinating and supporting multiple applications capable of adapting to specific computing needs
- Design and development of middleware software that is successful for certain types of applications (real time, sensor networks, etc.) is far from being easy
- In addition to responding to the challenges of these applications, it must respond to the demands imposed by the execution context (punctuality, real-time restrictions,...)

Introduction to Middleware
Introduction to SOA
Business process modeling
Web services composition

Middleware-based solutions "obsolescence"

Fundamental idea

Currently, the development of applications for specific domains has evolved towards paradigms that involve **encapsulating more knowledge and independence** of processing and interoperability **in the own software components** rather than in middleware, the paradigms called " **Service Oriented Architecture** " (SOA) and Agent Oriented Programming arouse more interest today

Introduction to Middleware
Introduction to SOA
Business process modeling
Web services composition

SOAP
XML
SW Programming

SOA characteristics

What's SOA?

A standard and flexible software architecture that unifies business processes by structuring big applications as a collection of "services"

These applications can be used by end users inside or outside the company

Architectural style

SOA is also an architectural style where all the functionalities, whether they are already at stake or are new, get grouped as services that communicate between themselves

Introduction to Middleware
Introduction to SOA
Business process modeling
Web services composition

SOAP
XML
SW Programming

Web Services

WS

are services implemented using a set of standards: SOAP, WSDL and UDDI, so that they can be accessed through a network or the Internet, and get executed remotely

Standard Protocol

Service Oriented Architecture Protocol (SOAP)

A common protocol to all the architecture's components

Services description language

Web Service Description Language (WSDL)

Discovery of available services

Universal Description Discovery and Integration (UDDI)

Services get classified into categories, according to the offering and how can be called

Introduction to Middleware
Introduction to SOA
Business process modeling
Web services composition

SOAP
XML
SW Programming

Fundamental

- We can talk about the first and the second generation of Web Services (WS)
- *Primitive* SOA corresponds to 1st Generation WS
- *Contemporary* SOA corresponds to 2nd Generation (WS 2.0)

Introduction to Middleware
Introduction to SOA
Business process modeling
Web services composition

SOAP
XML
SW Programming

WS implementation framework

Characterisation of *Base*-Web Services

- Have to be *abstract*:
  - Defined by an international standards (ISO, OASIS, ...)
  - Implemented on all platforms
- Constructive blocks, service and message descriptions based on WSDL
- Message exchanging component based on SOAP and its concepts
- UDDI-based service description registry, which sometimes includes service discovery
- A well-defined software architecture with messages, patterns and services composition

Introduction to Middleware
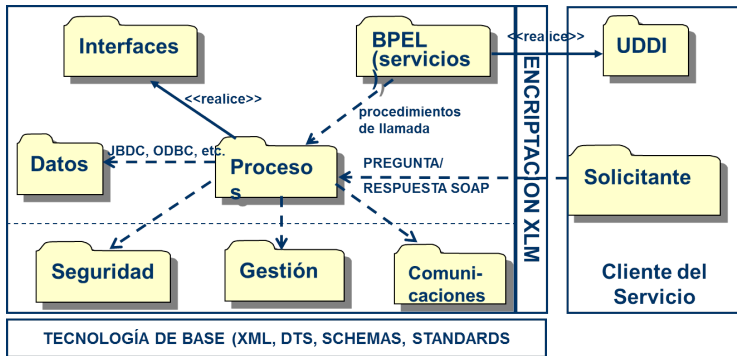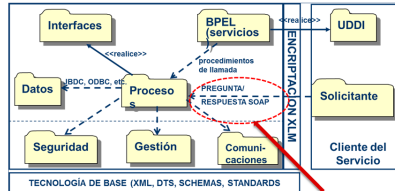**Introduction to SOA**
Business process modeling
Web services composition

SOAP
XML
SW Programming

Implementation of SW

Figure: Implementation of a WS architecture

Introduction to Middleware
Introduction to SOA
Business process modeling
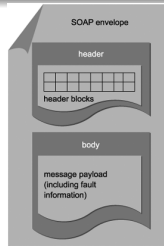Web services composition

SOAP
XML
SW Programming

SW and SOAP

Message passing vs. RPCs

- Problems that SOA implementation with RPCs/RMI shows
- RPC realisation based on XML documents (as autodescriptive messages)
- XML–RPC
- SOAP –successor of XML-RPC– allows remote objects calls with HTTP

Introduction to Middleware
Introduction to SOA
Business process modeling
Web services composition

SOAP
XML
SW Programming

SW and SOAP II
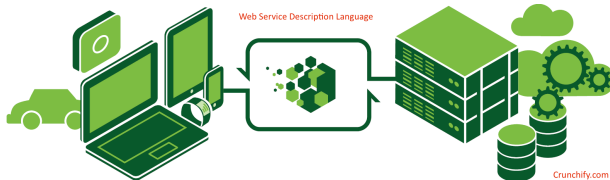

What SOAP gives us:

- To describe the (*"target"*) of a remote call
- Interoperable codification of data types within the 'remote-call' message
- To define relevant parts of each message
- Protocol independence (bindings instrumented with HTTP)

Introduction to Middleware
Introduction to SOA
Business process modeling
Web services composition

SOAP
XML
SW Programming

SOAP characteristics

Elements of the standard

- Namespaces for *wrappers*:
  http:\schemas.xmlsoap.org/soap/envelope/
- Namespaces for message contents codifying:
  http:\schemas.xmlsoap.org/soap/encoding
- Object references are allowed but only for objects that communicate through serialized communications (names used in messages are URLs)
- Passing of object references to a SOAP-receiver adapter
- Authentication of parts, information encryption, and compensation are not SOAP responsibilities

Introduction to Middleware
Introduction to SOA
Business process modeling
Web services composition

SOAP
XML
SW Programming

SW and XML

Web Service Description Language

Crunchify.com

WS

- Services are normally offered by a Web server
  - "XML-capable" servers have the greatest interest
- Then, Service Level Agreements ("SLA") are guaranteed

Introduction to Middleware
Introduction to SOA
Business process modeling
Web services composition

SOAP
XML
SW Programming

XML Web Services

Similarity with *software components*

- The client part has the same perception as if WS were software-components
- WS –'provided interfaces' are equivalent to the ones of software components

Relevant differences with software components

- *Required interfaces* will be never disclosed by a WS
- Required interfaces can only be used in the context in which they are offered, since WS are 'self-contained'
- Internet weak-coupling is always assummed between WS
- Interface binding-based reconfiguration banned

Introduction to Middleware
Introduction to SOA
Business process modeling
Web services composition

SOAP
XML
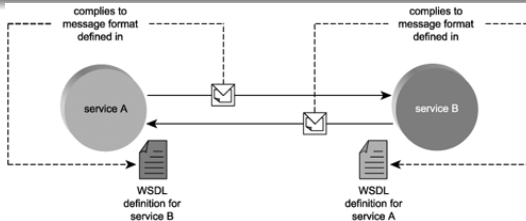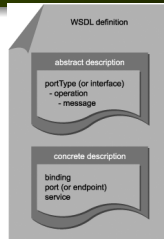SW Programming

SW Descriptions

Figure: Low coupling between services by using WSDL

Similarity with component descriptions:

- Software component's interfaces need metadata or a special annotation for their semi-formal description
- "*Web Services Description Language (WSDL)*" is the appropriate standard to solve the above issue
- Extensible framework definition

Introduction to Middleware
Introduction to SOA
Business process modeling
Web services composition

SOAP
XML
SW Programming

SW Descriptions with WSDL – II

- Service descriptions are fundamental for obtaining weak-coupled communication
- and also for getting a development infrastructure of software based on WS

- Abstract description
- Concrete description

Introduction to Middleware
Introduction to SOA
Business process modeling
Web services composition

SOAP
XML
SW Programming

Concrete description of services with WSDL

Objective:

Connects an abstract interface definition with a real technology that can be used for service implementation, i.e., for deployment into a given physical network's transport protocol environment

Concrete description structure

- *Binding*: sets the conditions needed to establish the "*physical*" connections of the service with the solicitors
- *Port*: "physical" direction through which any service can be accessed by using a specific protocol renaming, similar to the *end-point* concept in WSDL 2.0
- *Service*: groups of *end-points* that are the *correspondents* for providing a service, which is demanded at the other end

Introduction to Middleware
Introduction to SOA
Business process modeling
Web services composition

SOAP
XML
SW Programming

Abstract description of services with WSDL

Sets the characteristics of the service interface without any reference to its implementation technology

Abstract description structure

- *PortType* (WSDL 2.0 "Interface"): message ordering in groups of operations
- *Operation*: set of input/output messages ($\equiv$ operation's parameters)
- *Message* patterns that govern transmission of message sequences

Introduction to Middleware
Introduction to SOA
Business process modeling
Web services composition

SOAP
XML
SW Programming

Example, abstract WSDL description of a simple WS

```
1  <definitions name="StockQuote" targetNamespace="http://example.
       com/stockquote.wsdl"xmlns:tns="http://example.com/
       stockquote.wsdl"xmlns:xsd1="http://example.com/stockquote.
       xsd"xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"xmlns
       ="http://schemas.xmlsoap.org/wsdl/">
2    <types>
3     <schema targetNamespace="http://example.com/stockquote.xsd"
4               xmlns="http://www.w3.org/2000/10/XMLSchema">
5             <element name="TradePriceRequest">
6             <complexType>
7  <all><element name="tickerSymbol" type="string"/></all>
8               </complexType>
9             </element>
10            <element name="TradePrice">
11                <complexType>
12                    <all>
13                        <element name="price" type="float"/>
14                    </all>
15                </complexType>
16            </element>
17     </schema>
18    </types> ...
```

Introduction to Middleware
Introduction to SOA
Business process modeling
Web services composition

SOAP
XML
SW Programming

Example: <u>abstract</u> WSDL description of a simple WS –II

```
 1  ...
 2   <message name="GetLastTradePriceInput">
 3      <part name="body" element="xsd1:TradePriceRequest"/></
              message>
 4   <message name="GetLastTradePriceOutput">
 5      <part name="body" element="xsd1:TradePrice"/></message>
 6   <portType name="StockQuotePortType">
 7      <operation name="GetLastTradePrice">
 8        <input message="tns:GetLastTradePriceInput"/>
 9        <output message="tns:GetLastTradePriceOutput"/>
10      </operation>
11   </portType>
```

Introduction to Middleware
Introduction to SOA
Business process modeling
Web services composition

SOAP
XML
SW Programming

Example: <u>concrete</u> WSDL description of a simple WS

```xml
1   <binding name="StockQuoteSoapBinding" type="tns:
        StockQuotePortType">
2       <soap:binding style="document" transport="http://schemas.
            xmlsoap.org/soap/http"/>
3       <operation name="GetLastTradePrice">
4           <soap:operation soapAction="http://example.com/
                GetLastTradePrice"/>
5           <input><soap:body use="literal"/></input>
6           <output><soap:body use="literal"/></output>
7       </operation>
8   </binding>
9   <service name="StockQuoteService">
10      <documentation>My first service</documentation>
11      <port name="StockQuotePort" binding="tns:
            StockQuoteBinding">
12        <soap:address location="http://example.com/stockquote"
            />
13      </port>
14  </service>
15 </definitions>
```

[Introduction to Middleware](#)
Introduction to SOA
Business process modeling
Web services composition

SOAP
XML
SW Programming

WS protocol's stack

Discovery	UDDI	It's also a Web Service and works as one Web Services directory
Description	WSDL, WSFL/XLANG other for showing up	Given 1 or more WS, these notations describe the properties in a meta-level
Access	SOAP, SOAP with attachments	Given a Web Service instance, its access is based on messages
Transfer	HTTP,SMTP, other	Transfer SOAP messages, attachments included
Transport	TCP/IP, UDP, other	Transport data

Table: Fundamental WS-protocol stack

Introduction to Middleware
Introduction to SOA
Business process modeling
Web services composition

SOAP
XML
SW Programming

WS and programming models

- WSDL ports vs. component interfaces methods
- OOP model incompatibility
- Asynchronicity ($\neq$ RPC-based mechanisms)
- Round-trip call latency is unpredictable for WS requests
- Unpredictable delays caused by message storage in queues
- Concurrency constraints (caused by a great number of threads) in the client side
- Deployment of an asynchronous model for WS requests, similar to the one based on message queues

Introduction to Middleware
Introduction to SOA
Business process modeling
Web services composition

SOAP
XML
SW Programming

WS-I

Web Services Interoperability Interface

- Industrial consortium initiated for promoting *interoperability* amongst the stack of WS specifications
- "*Guidelines*" definition isn't any new standard
- Recently, became part of OASIS
- http://ws-i.org/Profiles/BasicProfile-1.2-2010-11-09.html
- *Self-certification* guidelines inclusion

Introduction to Middleware
Introduction to SOA
Business process modeling
Web services composition

SOAP
XML
SW Programming

WS-I

Web Services Interoperability Interface

WS▸I
WEB SERVICES
INTEROPERABILITY
ORGANIZATION

Basic Profile Version 1.2

Final Material

2010-11-09

This version:
http://www.ws-i.org/Profiles/BasicProfile-1.2-2010-11-09.html
Latest version:
http://www.ws-i.org/Profiles/BasicProfile-1.2.html
Errata for this Version:
http://www.ws-i.org/Profiles/BasicProfile-1.2-errata-2010-11-09.html
Editors:
    Robert Chumbley , IBM
    Jacques Durand , Fujitsu
    Micah Hainline , Asynchrony Solutions
    Gilbert Pilz , Oracle
    Tom Rutt , Fujitsu
Administrative contact:
    secretary@ws-i.org

Abstract

This document defines the WS-I Basic Profile 1.2, consisting of a set of non-proprietary Web services specifications, along with clarifications, refinements, interpretations and amplifications of those specifications which promote interoperability. It also conta[...] set of executable test assertions for assessing the conformance to the profile.

Status of this Document

This is a final specification. Please refer to the errata, which may include normative corrections to it.

Introduction to Middleware
Introduction to SOA
Business process modeling
Web services composition

SOAP
XML
SW Programming

WS-II

Web Services Interoperability Interface

Table of Contents

Introduction to Middleware
Introduction to SOA
Business process modeling
Web services composition

SOAP
XML
SW Programming

SW and software components models standards

Current proposals in the software components market

- CORBA, is in the market for a long time: business-oriented computation and legacy applications
- Java (prior Sun Systems) of Oracle: the most important language/platform for application and Web servers development
- COM+ and .NET of Microsoft: takes advantage of the quasi-"monopolistic" Windows operating system within the low-cost laptop market
- The new models of WS have become a new software layer, which are currently essential in the business model, with a software integration potential unprecedented!

Introduction to Middleware
Introduction to SOA
Business process modeling
Web services composition

Business Processes characteristics

Business modeling

"Business Modeling *buzzword*"

- Dissemination:
  - More than 5.9M of links to English written documents,
  - More than 1.1M of links to Spanish written documents
- In specialized digital libraries :
  - *ACM digital library:* 20,000 hits to "business process modeling"
  - *IEEE digital library:* 750 hits to "business modeling"

Introduction to Middleware
Introduction to SOA
Business process modeling
Web services composition

Business Processes characteristics

Business modeling-II

Business Model (BM)

Abstraction about how a business works. It depends on the expert(s) who made it.

Simplified view of the business structure to spread or improve it. It is considered part of the definition of the information system's requirements, from which the information system will take place

Introduction to Middleware
Introduction to SOA
Business process modeling
Web services composition

Business Processes characteristics

Business modeling-III

Business

Set of activities of a company that owns or uses resources, have several goals and if it is carried out correctly and creates value (usually economic) for the company

Introduction to Middleware
Introduction to SOA
Business process modeling
Web services composition

Business Processes characteristics

Motivation of using BP technology

Benefits

- BBPP are the most valuable asset of a company
- Propitiates a better understanding of the key mechanisms of a business or company
- Acts as the basis for the creation of appropriate information systems that support the business
- Supports the improvement of the structure and operation of the current business
- Shows the structure of an innovative business
- Experiments with new business concepts or copy or study a concept used by a competing company
- Identifies outsourcing opportunities for products or supplies

Introduction to Middleware
Introduction to SOA
Business process modeling
Web services composition

Business Processes characteristics

Characteristics

Business processes:

- Very large and complex
- Highly dynamic
- Distribution
- Depends on economic activity or on the company's business rules
- Partially automatable

Introduction to Middleware
Introduction to SOA
Business process modeling
Web services composition

Business Processes characteristics

Views

Objective:

They allow highlighting only the important information, at a certain stage, to describe more easily the functions performed by a business process

Types:

- Product/services
- Data
- Process
- Function
- Organizational

Introduction to Middleware
Introduction to SOA
Business process modeling
Web services composition

Business Processes characteristics

Differences with other process types

Types of processes	Industrial	Information	Business
Focus	Things	Data	Relationships
Purpose	Transform and assembly materials and components in other components and final products using resources	Process and transmit structured data and non- structured and knowledge	Meet conditions that satisfy the needs of participants clients or users
Character ization	Traditions of Industrial Engineering	Traditions of Informatics and Computer Engineering	Based on structures of communication and coordination of humans through languages
Actions	Assembly Transform Transport Store Inspect	Send, Invoke Record, Recuperate Consult Classify	Solicit, Promise Offer, Turn down Propose, Cancel Measure

Introduction to Middleware
Introduction to SOA
Business process modeling
Web services composition

Business Processes characteristics

Business Process based Systems

Development elements

- BPS are based on information technologies (IT)
- Manage all phases of the business process life cycle
- Transparent deployment of computer systems
- Provide visibility and control
- Integrate ICT technologies as they have reached a level of maturity
- Process-guided management
- They allow to establish a common language among all the stakeholders

Introduction to Middleware
Introduction to SOA
Business process modeling
Web services composition

Business Processes characteristics

Structure of Business Process Modeling (BPM)

Process schemata

- More general/abstract perspective than carrying out system development based on OO/WS
- Generalize and integrate development processes focused on <u>data</u> and functional decomposition

OO/WS (before)	**BPM** (now)
Applications	Manufacturing processes tools & services
Application server	Processes
Data Base	Fundamental process

Introduction to Middleware
Introduction to SOA
Business process modeling
Web services composition

Business Processes characteristics

Relationship between BPM and WS

Standard BPM stack		
Business Process Modeling Notation (BPMN)		
Semantic Business Process Model (SBPM)		
Business Process eXtension Layers (BPXL)		
Web Services stack (WSDL,UDDI, etc.)		
WS-CDL	BPEL	BPQL

Introduction to Middleware
Introduction to SOA
Business process modeling
Web services composition

Orchestration
Choreography

Ontology of a service-oriented architecture WS 2.0
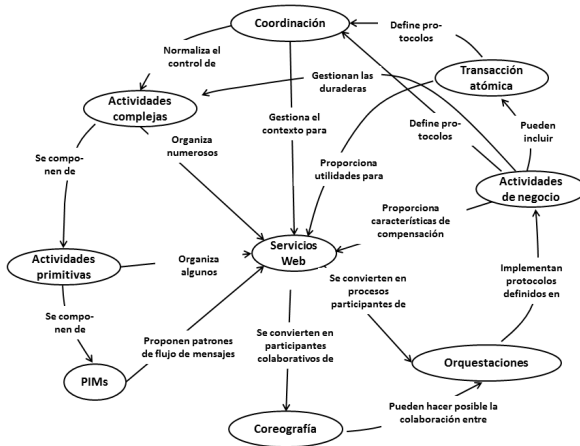
Figure: Concepts and relations that a WS 2.0 architecture includes

Introduction to Middleware
Introduction to SOA
Business process modeling
Web services composition

Orchestration
Choreography

Introduction to WS composition

Figure: WS collaboration

WS composition

"Process of aggregation of multiple services into a single service in order to carry out more complex functions by an application or a business model"

Introduction to Middleware
Introduction to SOA
Business process modeling
Web services composition

Orchestration
Choreography

General Ontology of services composition in WS 2.0
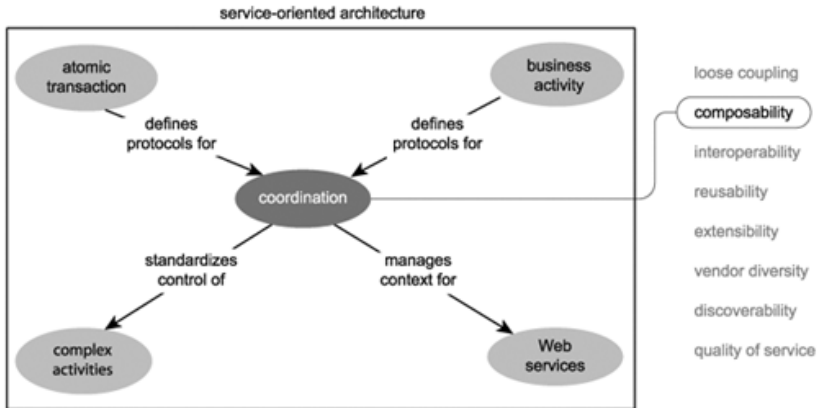
Figure: Relationship with the other parts of the architecture

Introduction to Middleware
Introduction to SOA
Business process modeling
Web services composition

Orchestration
Choreography
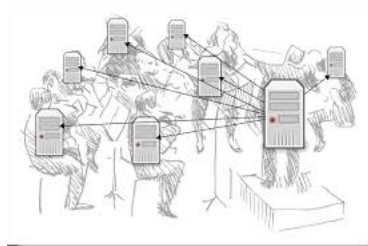
WS Orchestration

Figure: Orchestration of WS

Fundamental idea

It is a well-defined and centralised part of a business information system *work-flow logic*, which eases the *interoperability* among 2 or more different applications

Introduction to Middleware
Introduction to SOA
Business process modeling
Web services composition

Orchestration
Choreography

WS orchestration

Objetives

- To make true a possible integration of big business processes without the necessity of rebuilding applications
- Application collaboration is obtained through the introduction of a *work-flow logic* in business models
- To abstract the *work-flow logic* from the problem solution
- Orchestration's deployment becomes most important in service oriented development
  - *work-flow logic* of a business process then only needs requests of previously deployed services

Introduction to Middleware
Introduction to SOA
Business process modeling
Web services composition

Orchestration
Choreography

Orchestration as a service
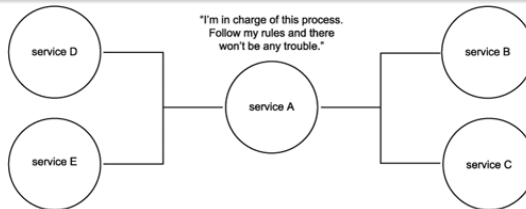
- Company federation and service orientation
- Service design propitiates the *interoperability*
- As it is another service, it gives system's extension without affecting any of the component interoperability
- The logics that is included in an orchestration allows us to obtain the representation of any company as standard models

Introduction to Middleware
Introduction to SOA
Business process modeling
Web services composition

Orchestration
Choreography

Web Services Business Process Execution Language (WS-BPEL)

Característics of WS-BPEL/BPEL4WS/BPEL

- Key extension for making possible to obtain 2nd generation WS
- WS-BPEL makes use of concepts and WS-* terminology, in order to model business processes
- Currently valid standard: WS-BPEL 2.0 (see www.oasis-open.org)

Introduction to Middleware
Introduction to SOA
Business process modeling
Web services composition

Orchestration
Choreography

Web Services Interoperability Interface Information at OASIS

Web Services Interoperability Interface

Introduction to Middleware
Introduction to SOA
Business process modeling
Web services composition

Orchestration
Choreography

Business protocols and process definition

Orchestration characteristics

- *Work-flow logics*: events, conditions and business rules
- A protocol is defined, i.e.: how the participants of an orchestration interact with each other in order to perform each one a task of the business
- The *work-flow logics* in an orchestration is contained in a business process definition

Introduction to Middleware
Introduction to SOA
Business process modeling
Web services composition

Orchestration
Choreography

Services of processes and among associated
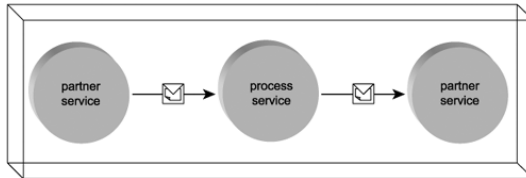
Figure: A process service initially called by an associated service and then invoked by another

Orchestration semantics

- The participant processes are represented as services
- The *process service* can get being associated (become a "partner") and/or being called remotely

Introduction to Middleware
Introduction to SOA
Business process modeling
Web services composition

Orchestration
Choreography

Work-flow descriptions with WS-BPEL

Fundamental idea that supports BPEL

- To decompose any *work-flow logics* into a series of pre-established basic actions
- To identify *work-flow* basic actions that can be assembled by using a logics of structured activities: *secuence*, *switch*, *repetition*, etc.
- It is important to get an unambigous and previously defined execution order of activities

Introduction to Middleware
Introduction to SOA
Business process modeling
Web services composition

Orchestration
Choreography

Work-flow descriptions with WS-BPEL - II

Orchestration and synchronization

- A work flow cannot end until all the activities that it contains are finished
- A global synchronization/orchestration logics is therefore obtained from individual *work-flows*
- The modularity principle is then applied to *work-flows*, which can be obtained by the definition of *links*: formal dependencies among the activities carried out on different flows
  - An activity cannot be completed until all the requirements defined on its output links are fulfilled
  - Before starting any activity the input links requirements (contained items) must be satisfied

Introduction to Middleware
Introduction to SOA
Business process modeling
Web services composition

Orchestration
Choreography

Orchestration and SOA
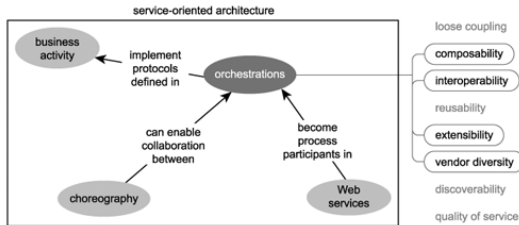
Figure: Orchestration connected with other parts of a SOA

Business companies automatization principles

- The process logics of a company becomes centralised as an orchestration
- Orchestrations propitiate application environments oriented to extensible and adaptative services

Introduction to Middleware
Introduction to SOA
Business process modeling
Web services composition

Orchestration
Choreography

Benefits of using orchestration in a SOA

Summary

- Centralised modification of the *work-flow logics*
- To ease the fusion of business processes
- Wide scale service definition (SOAs) leverages the future evolution towards federated companies
- Current technology helps to attain the objective: today's middleware allows us to integrate orchestration engines in environments of service oriented applications

Introduction to Middleware
Introduction to SOA
Business process modeling
Web services composition

Orchestration
Choreography
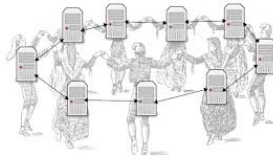
WS choreography

Figure: Choreographies of WS

Fundamental idea

A well-defined part of the *business logics* that eases the *interoperability* of services when collaboration goes beyond of company/organisation limits.

Can be seen as universal collaboration and interoperability patterns for carrying out business tasks that are shared by different organisations.

Introduction to Middleware
Introduction to SOA
Business process modeling
Web services composition

Orchestration
Choreography
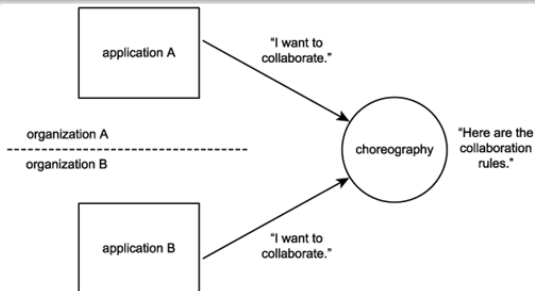
WS choreography – II

Figure: A choreography makes possible the colaboration among participants.

Motivation

Multiple services of different organisations need a "choreography" when they have to work together in order to attain a common objective in the deployment of a business task

Introduction to Middleware
Introduction to SOA
Business process modeling
Web services composition

Orchestration
Choreography

Choreography as a service

Characteristics of a choreography

- Serve for exchanging *public messages*, i.e., to set the collaboration among WS that belong to different organisations
- Differently from the "orchestration" mechanism, no organisation can control the *collaboration logics*
- Can be also used for creating a collaboration among applications that belong to the same organisation
- *Actions* within a choreography are structured as message exchange sequences among WS

Introduction to Middleware
Introduction to SOA
Business process modeling
Web services composition

Orchestration
Choreography

Choreography as a service – II

Channels

- A message exchange is defined as a relationship between one pair of roles, each one of these assumed by a partner
- A choreography action is defined by a set of role-pairs of a WS
- A channel defines the representation of messages exchange between 2 partners of a role-pair
- Information about a channel can be passed on a message in order to obtain a way to implement *service discovery*
- *Work units* of a choreography include interactions among WS, which are constrained by conditions set to finish up each communication successfully

Introduction to Middleware
Introduction to SOA
Business process modeling
Web services composition

Orchestration
Choreography

WS-CDL choreography description

Web Services Choreography Description Language

- Programming language based on XML that serves to describe the collaboration among partners by definition of each relation's participant common and observable behaviors

- The language specification is due to W3C (W3C Web Services Choreography Working Group, closed in July 2009), ver: https://www.w3.org/TR/ws-cdl-10/

- At moment, we can observe overlapings between WS-CDL and WS-BPEL for orchestrations

- WS-CDL specifications propitiate dynamic discovery of services and the widescale collaboration among many participants

Introduction to Middleware
Introduction to SOA
Business process modeling
Web services composition

Orchestration
Choreography

Web Services Choreography Description Language Specification Site

WS-CDL Interface

W3C

Web Services Choreography Description Language Version 1.0

W3C Candidate Recommendation 9 November 2005

This version:
http://www.w3.org/TR/2005/CR-ws-cdl-10-20051109/
Latest version:
http://www.w3.org/TR/ws-cdl-10/
Previous version:
http://www.w3.org/TR/2004/WD-ws-cdl-10-20041217/
Editors:
Nickolas Kavantzas, Oracle
David Burdett, Commerce One
Gregory Ritzinger, Novell
Tony Fletcher, Choreology
Yves Lafon, W3C
Charlton Barreto, Adobe Systems Incorporated

Copyright © 2005 W3C® (MIT, ERCIM, Keio). All Rights Reserved. W3C liability, trademark and document use rules apply.

Abstract

The Web Services Choreography Description Language (WS-CDL) is an XML-based language that describes peer-to-peer collaborations of participants by defining, from a global viewpoint, their common and complementary observable behavior; where ordered message exchanges result in accomplishing a common business goal.

The Web Services specifications offer a communication bridge between the heterogeneous computational environments used to develop and host applications. The future of E-Business applications requires the ability to perform long-lived, peer-to-peer collaborations between the participating services, within or across the trusted domains of an organization.

The Web Services Choreography specification is targeted for composing interoperable, peer-to-peer collaborations between any type of participant regardless of the supporting platform or programming model used by the implementation of the hosting environment.

Status of this Document

This section describes the status of this document at the time of its publication. Other documents may supersede this document. A list of current W3C publications and the latest revision of this technical report can be found in the W3C technical reports index at http://www.w3.org/TR/.

This is the 9 November 2005 W3C Candidate Recommendation of "Web Services Choreography Description Language Version 1.0". W3C publishes a technical report as a Candidate Recommendation to indicate that the document is believed to be stable

W3C Candidate Recommendation

Introduction to Middleware
Introduction to SOA
Business process modeling
Web services composition

Orchestration
Choreography

Web Services Choreography Description Language Specification Contents

WS-CDL Interface

Publication as a Candidate Recommendation does not imply endorsement by the W3C Membership. This is a draft document and may be updated, replaced or obsoleted by other documents at any time. It is inappropriate to cite this document as other than work in progress.

W3C Candidate Recommendation

Table of Contents

Introduction to Middleware
Introduction to SOA
Business process modeling
Web services composition

Orchestration
Choreography

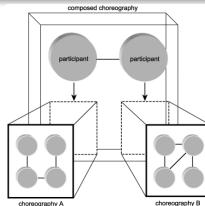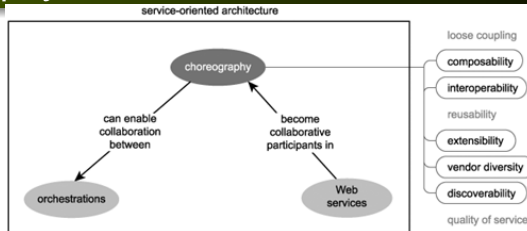# Reusability, compositionality and modularity

Figure: A choreography made up of 2 smaller choreographies.

- They are designed in a *reusable* way to be applied to different business tasks, which share actions
- Propitiates assembling of choreographies through an importation service inside a framework
- They can be structured into modules, which include taks to be used by a hierarchy of choreographies

Introduction to Middleware
Introduction to SOA
Business process modeling
Web services composition

Orchestration
Choreography

Choreography and SOA

Fundamental ideas

- Choreographies allow extending or dynamically modifying the participant business processes
- Help to configure complex SOAs that can go beyond the company limits
- Propitiate WS discovery and agil design and deployment of business tasks inside any organisation

Introduction to Middleware
Introduction to SOA
Business process modeling
Web services composition

Orchestration
Choreography

Orchestrations vs. choreograhies

Coincidences and diferences

- An orchestration can be understood as the specific application of a choreography to a given business
- Either orchestrations as choreographies can be used for modeling complex message exchange patterns
- An orchestration normally represents the work-flow of an organisation:
    - The company owns and controls the collaboration logics
- The control logics of a choreography does not usually belong to only one organisation:
    - the message exchange pattern proposed by one choreography is normaly used for collaboration among WS of different organisations

Introduction to Middleware
Introduction to SOA
Business process modeling
Web services composition

Orchestration
Choreography

Cites

[   ]

Introduction to Middleware
Introduction to SOA
Business process modeling
Web services composition

Orchestration
Choreography

References I

Bibliographic references #26,27,29,33,35,52,69,77,78,93 y 126 of the book:

Capel, M. (2016).

Desarrollo de software ys sistemas basados en componentes y servicios.

Garceta, **Básica**.