IS 651: Distributed Systems
Chapter 6: WS-*

Jianwu Wang

Fall 2020

Learning Outcomes

- After learning chapter 6, you should be able to
  - Know the reason for additional WS specifications
  - Know the differences between orchestration and choreography
  - Build server-side XSLT and know its differences from client-side XSLT

Web Services Interoperability (WS-I) basic profile

- XML 1.0
- XMLSchema 1.0
- SOAP 1.1
- WSDL 1.1
- UDDI 2.0

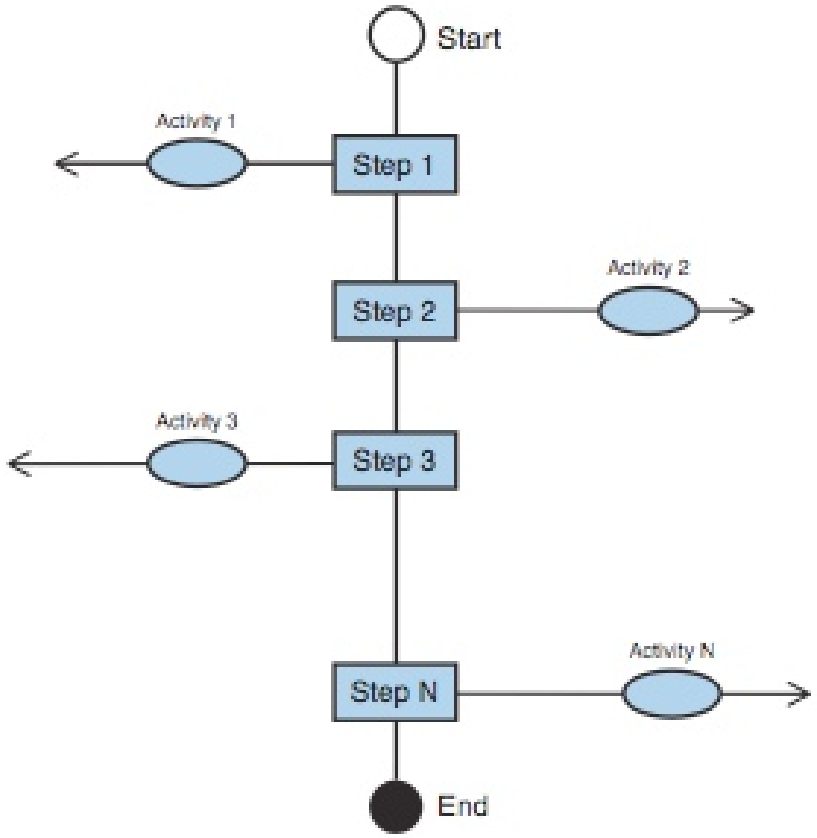
WS-*

- Complex information systems require many more non-functional services that support the functional ones
- Another main requirement is service composition
- These include workflow, management, security, reliability, transactions, and quality of service (QoS)
- A great number of web service specifications have arisen to offer these kinds of services called WS-*

Process	WS-BPEL, CDL		
QoS	WS-Security, WS-ReliableMessaging, and Addressing		
Discovery	UDDI (and others)		
Description	WSDL		
Messaging	SOAP		
Data	XML	XMLSchema	Namespaces
Transport	HTTP	URI	Unicode

Orchestration and Choreography

- Deals with business process and workflow
- Orchestration is intra-organization
- Choreography is inter-organization
- Very important examples of WS-*

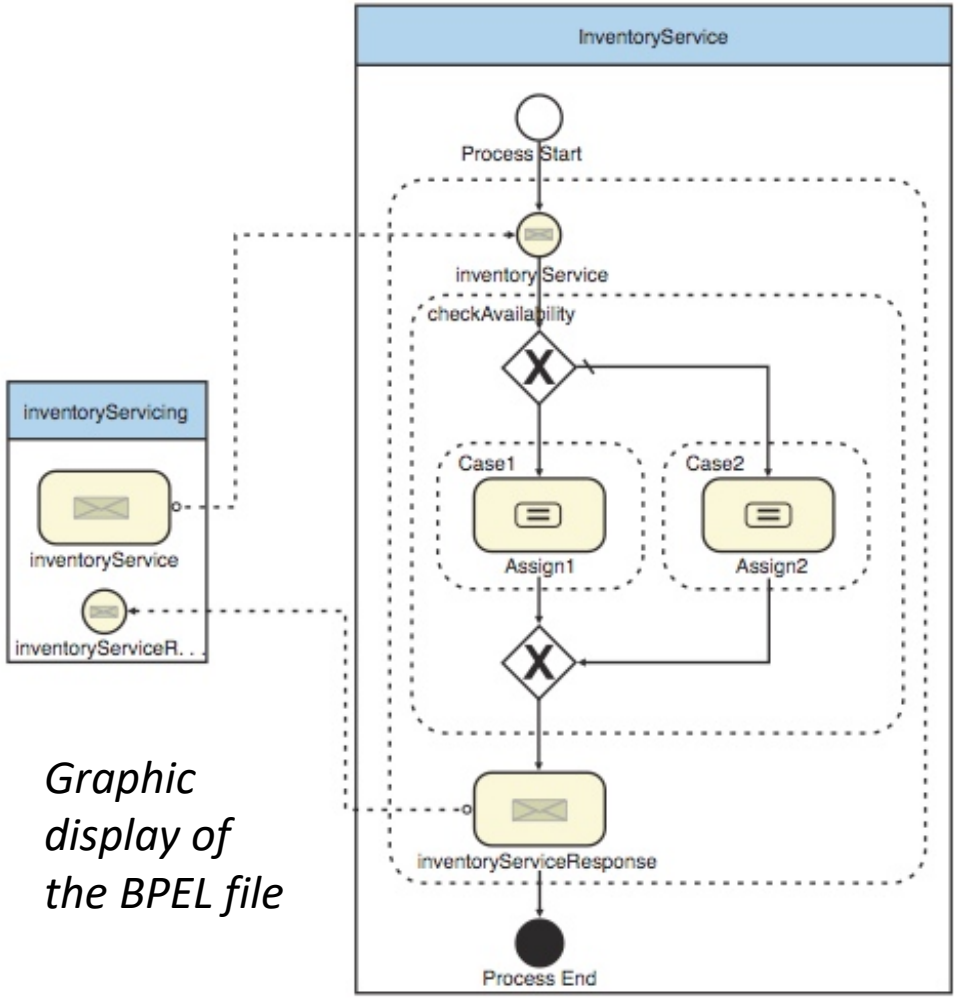


Orchestration

- Any business process can be broken down into several steps. Each step can be further broken down into a series of business activities.
- WS-BPEL specification provides an elegant means to express these activities with the help of a set of pre-defined XML elements
  - *Declaration elements: process, variable, partnerLink.*
  - *Activity elements for processes: **assign**, throw, wait, empty, ...*
  - *Activity elements for services: **invoke**, reply, wait, receive, ...*
  - *Structured Activity elements: while, pick, flow, sequence, ...*
  - *…*
- A BPEL process can be exposed as a service by defining its WSDL and linking to it
  - An important feature for hierarchical service composition

Orchestration in Graphic User Interface (GUI)

BPEL source code:
http://userpages.umbc.edu/~jianwu/is651/programs/ch6/bpel.xml



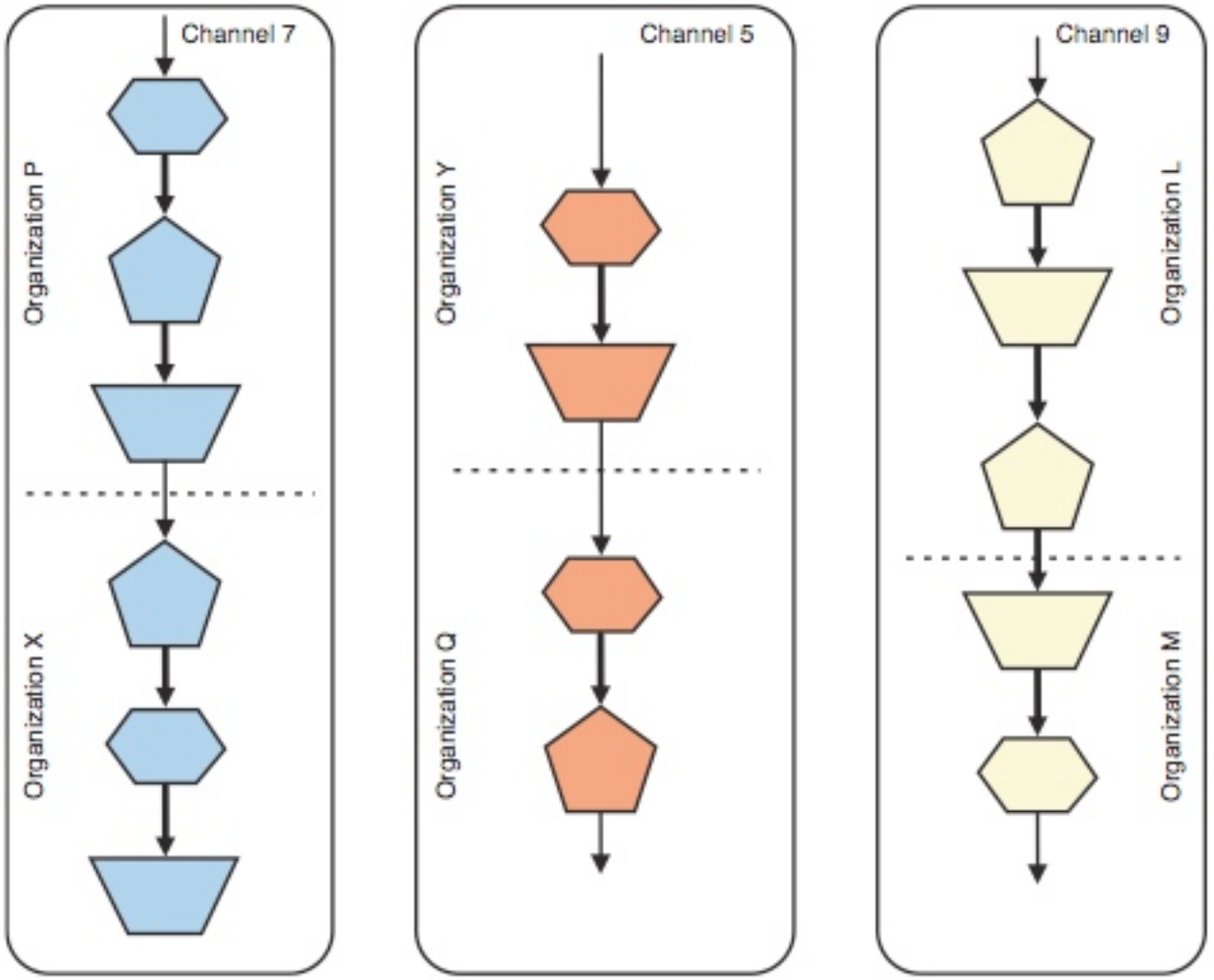
Graphic display of the BPEL file

BPEL Execution Environment

- A BPEL execution environment contains a web service framework and a BPEL server for executing the process instances
  - Like classes in object-oriented programming, a process can also have many instances
- The BPEL execution environment provides all the necessary life-cycle requirements, such as instantiation, transaction management, and so on, as per the WS-BPEL specifications
- Typically a BPEL execution environment works in combination with a web service environment such as Java Platform and/or .NET environment

Choreography

- WS-CDL specification is designed to promote the business interaction and information interchange among multiple cooperating business organizations



Server-side XSLT

- We saw client-side XSLT in Chapter 5, but most XSLT processing is done on the server using dedicated software library
- You will learn how to use SAXON (http://saxon.sourceforge.net/), a java-based XSLT processor, to do the transformation by doing the homework

Server-side XSLT Example

- XML file: http://userpages.umbc.edu/~jianwu/is651/programs/ch6/book_sample.xml
- XSLT output: http://userpages.umbc.edu/~jianwu/is651/programs/ch6/book_sample.xhtml

XSLT in Web Services

- XSLT is also used to transform XML documents in web services from one XML format to another when the participants need it.
- In the PO example we used earlier, one participant might use a different XMLSchema for the PO natively on their system than the one that is used by the web service that processes the POs even though all the actual content is the same