Orchestration, Choreography, and Collaboration

Carol McDonald

Technology Evangelist

Sun Microsystems

Sun microsystems

We make the net work.

Agenda

- Orchestration & Choreography
  - What is Orchestration/Choregraphy
  - Why
  - How Examples
- Collaboration
  - What is collaboration
  - Why
  - How Examples

Note

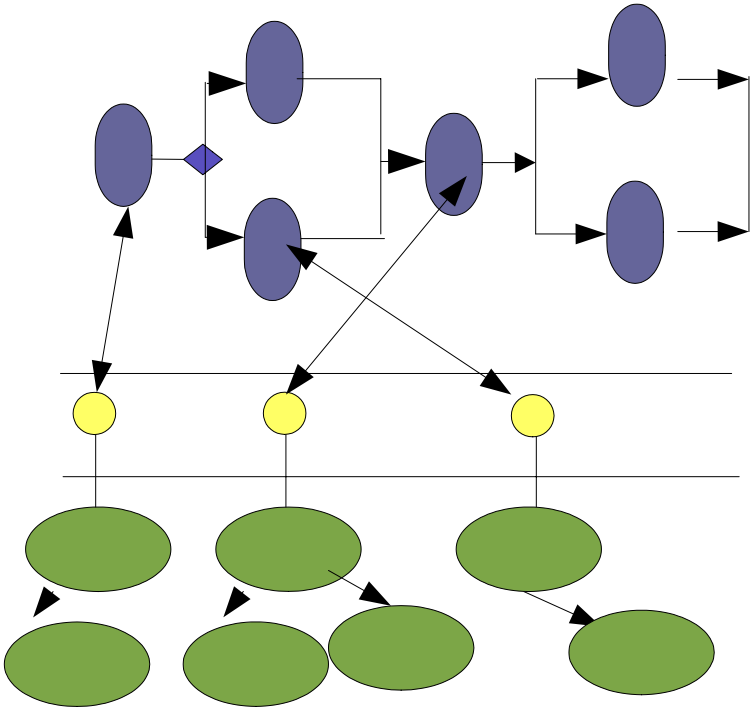
- even though Carol Mcdonald is a full time employee of Sun the contents here are created as her own personal endeavor and do not reflect any official stance by Sun Microsystems

What is Orchestration / Choreography?

What is Orchestration, choreography?

- Both related to composing web services for building dynamic, flexible business processes.
- Orchestration defines the interactions and process flow among Web services in a business process
- Choreography defines the flow of information exchanges among a set of participants to implement a business Process composing multiple web services
- Choreography is generally more public view but distinction is blurred, and the two viewpoints are converging

What is Orchestration?



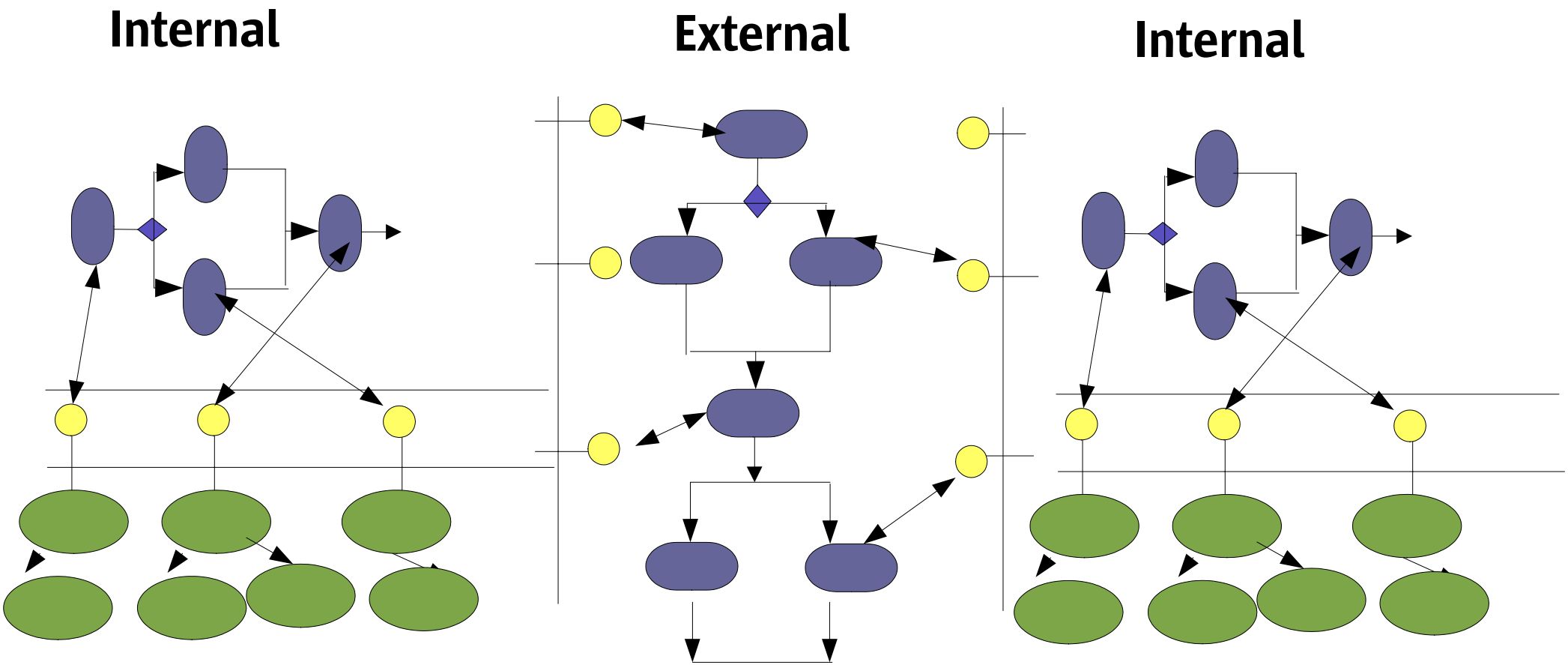
Orchestration

SOAP messaging

WSDL Interfaces

Components

What is Choreography?



Why Orchestration / Choreography?

Why?

- A "stack" of layers has been specified for the interoperability of Web Services.
- Additional layers are needed in order to enable Web Service composition.

Choreography	automated business process require a defined choreography of messages to compose web services
WSDL	WSDL describes the static interface of a Web Service. It defines the protocol and the messaging of end points
Soap	SOAP defines Platform independent XML message format and exchange

Why ?

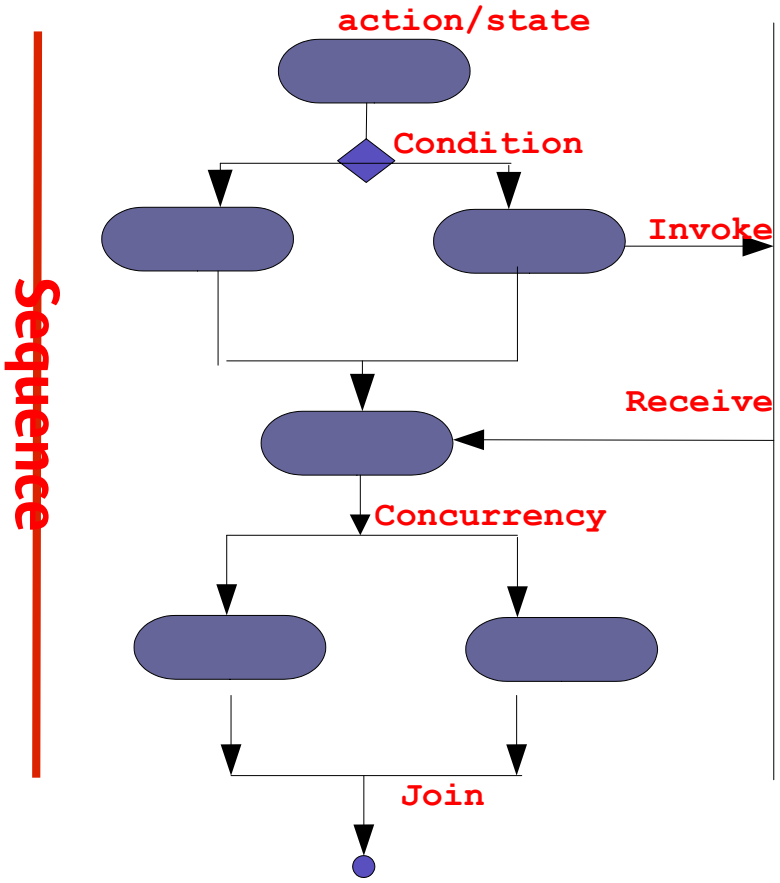
- Business Processes need to integrate and interact with WebServices
- Strong business process automation initiatives

Why?

- There is a need for describing complex interactions between web services:
  - Can messages be sent and/or received in any order?
  - What rules govern sequencing of messages?
  - Is there any relation among any incoming and/or outgoing messages?
  - Is there a "start" and an "end" of a given sequence? Can a given sequence be partially "undone"?
  - Can a global view of the overall exchange of messages be drawn?

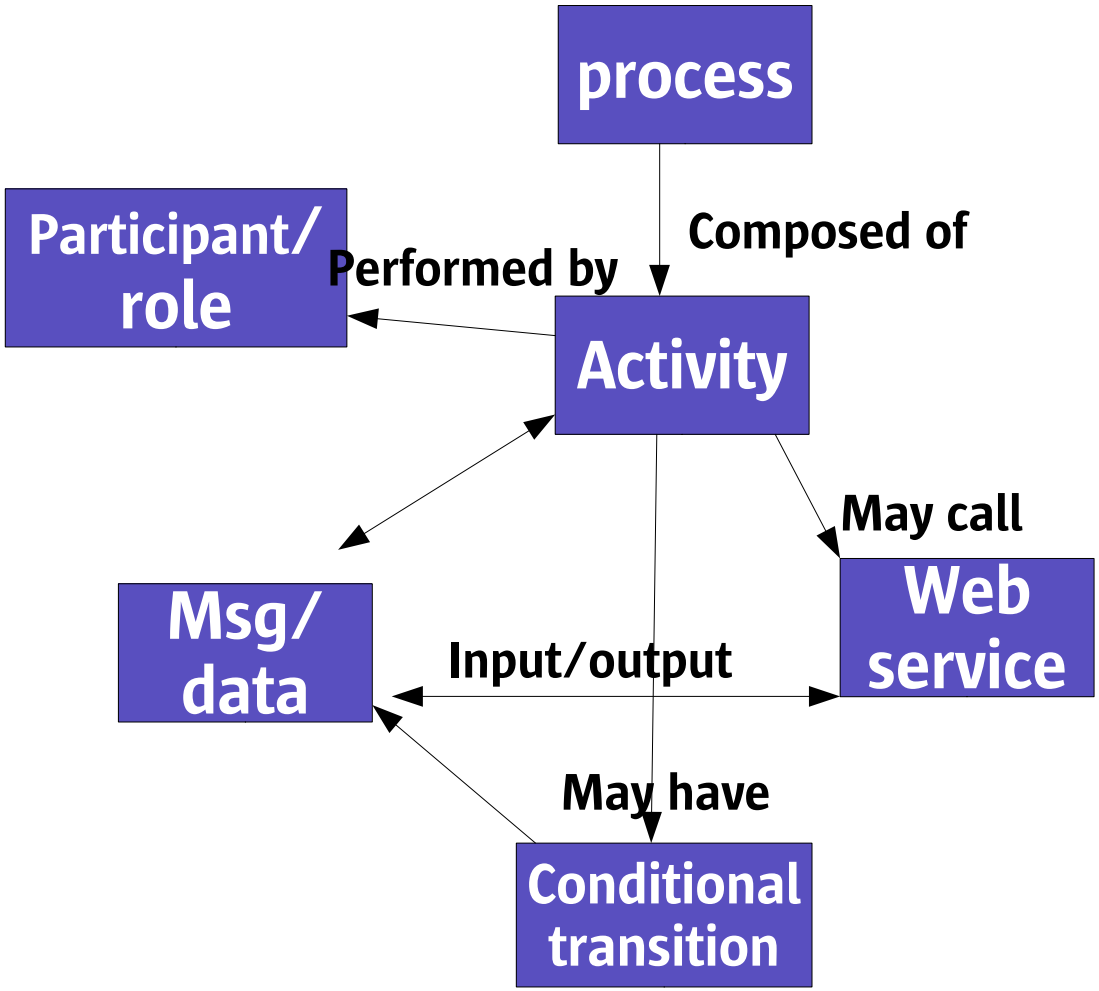
How Orchestration / Choreography?

General Characteristics of Orchestration/Choreography

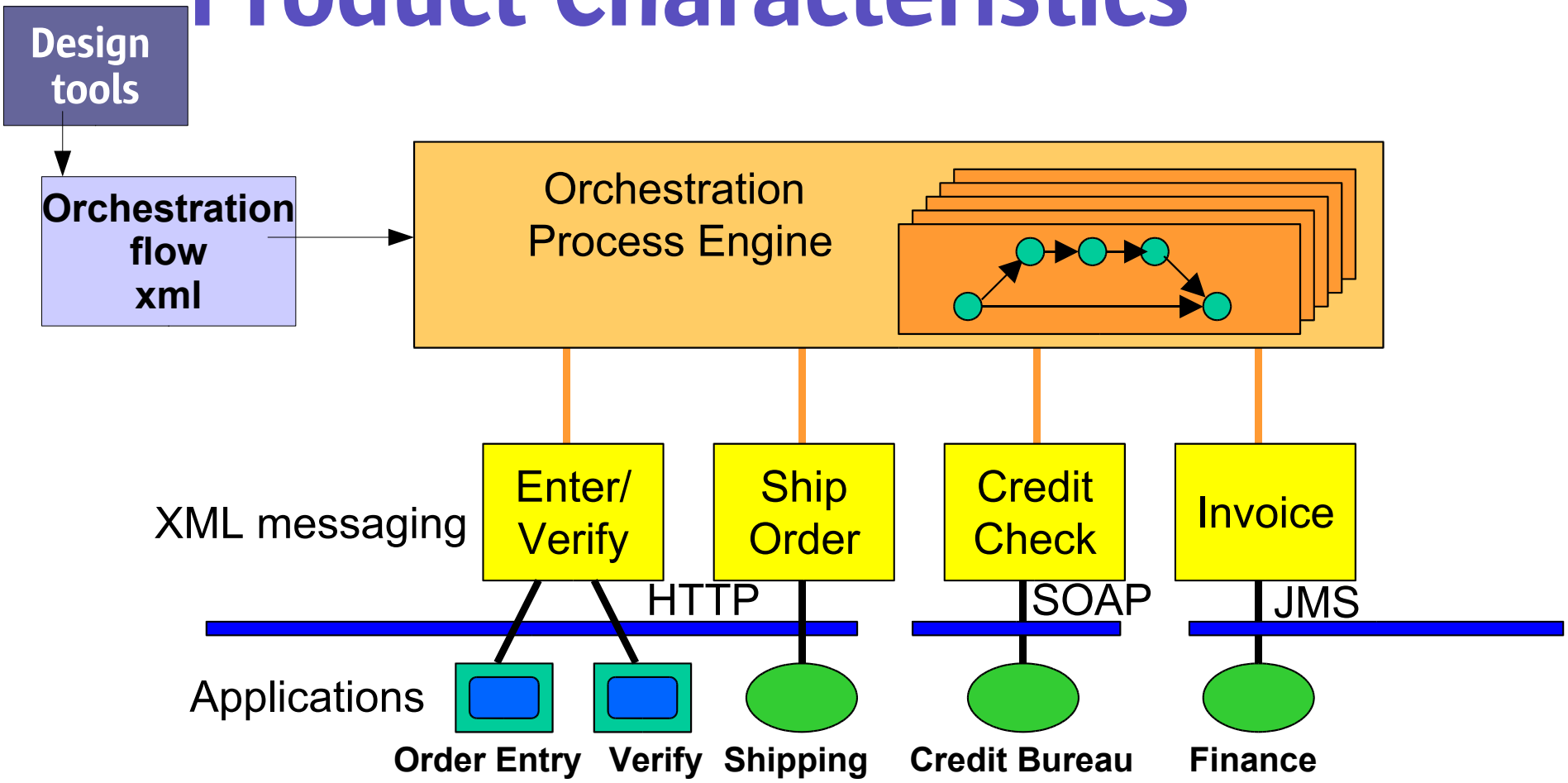


- Support for specifying states/ actions, events, control flow
- Action often mapped to wsdl port
- Flow: sequential, conditional, concurrent
- Based on mathematical pi-calculus
- Call or Receive from other Web Services
- Partners, roles
- Exceptions, transaction compensation
- Xml messaging, correlation

Meta-Model


process
Composed of
Participant/ role
Performed by
Activity
May call
Msg/ data
Input/output
Web service
May have
Conditional transition

Orchestration/Choreography Product Characteristics



Orchestration /Choreography Examples

BPML (from bpmi.org)

- XML syntax and abstract model for specifying executable business processes
- Process in BPML
  - Based on message exchange
  - Composed of activities
    - control flow (sequence, switch, for each, join...)
- Participants
  - Participate in a process by consuming and producing messages
  - Can be web services or other nested processes

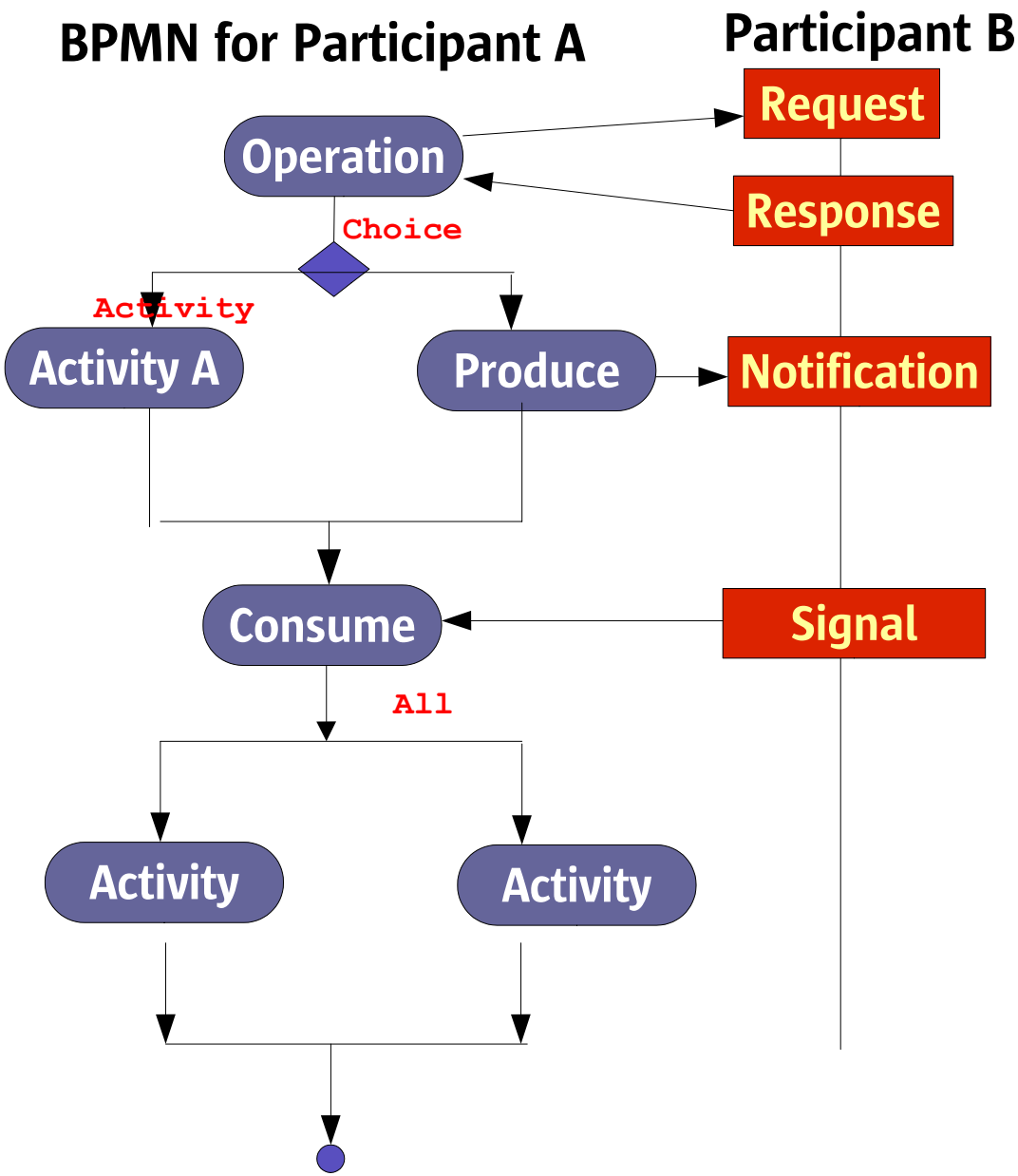
ref: http://www.bpmi.org

BPML

- Business Process Modeling Language
  - developed by Intalio, Sterling, Sun, CSC
  - meta-language for describing business processes language executed by a BPMS system
- Key features
  - basic activities for sending, receiving, and invoking services
  - handles conditional, sequential, and parallel activities
  - persistence, correlation, and composition support, transactions, exception handling

Ref http://www.bpn

BPML example



BPML for Participant A

```
<process name="processName">
  <sequence>
    <operation name="operation">
      <participant name="partB">
        <output message = "request"/>
        <input message = "response"/>
    </operation>
    <choice>
      <event...>
        <action ....>
        <action  ...>
    </choice>
    <all>
       <activity ...>
       <activity ...>
    </all>
  </sequence>
</process>
```

Re: http://www.bpmi.org

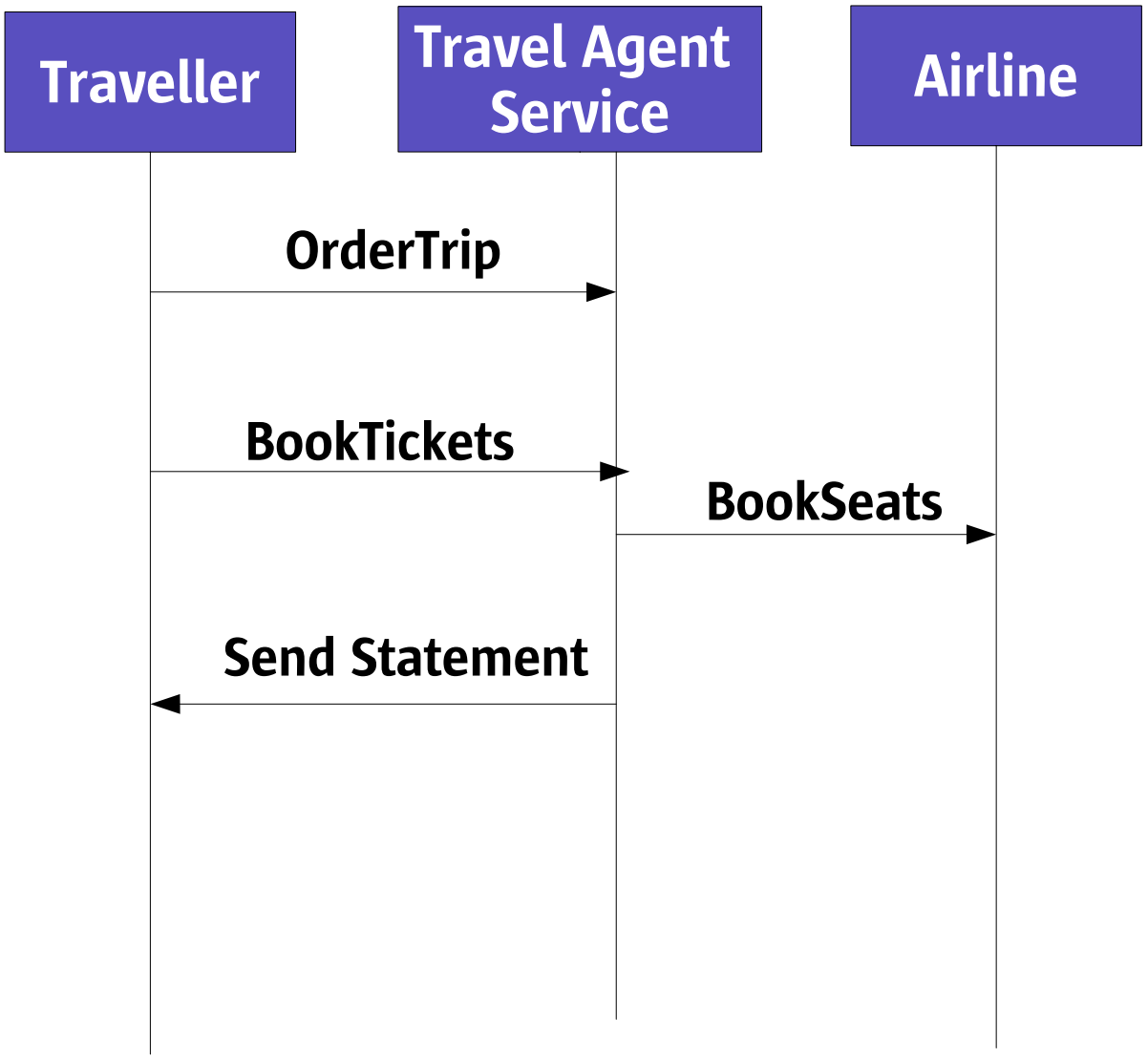
WSCI (from BEA, Intallio, Sun)

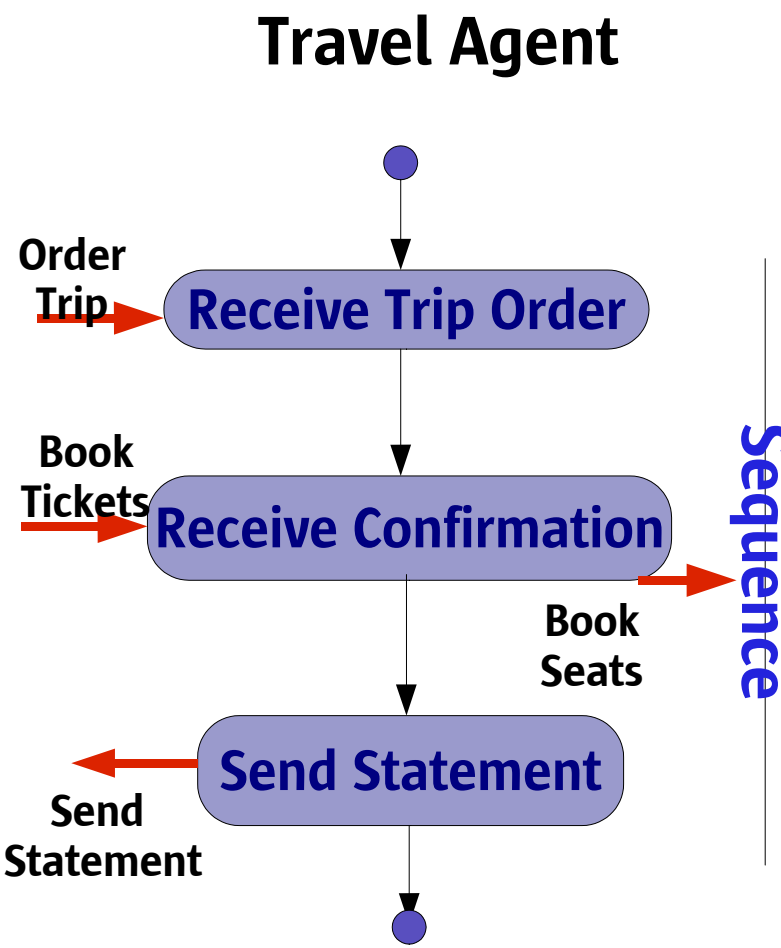
- describes how Web Service operations can be choreographed in the context of a message exchange in which the Web Service participates
- dynamic interface of a service in the context of a particular process
  - WSCI interface describes the order in which messages can be sent or received in a given message exchange,
  - the rules which govern such ordering,
  - The boundaries of a message exchange

WSCI

- WSCI supports:
  - Message choreography
  - Transaction boundaries and compensation
  - Exception handling
  - Thread management
  - Properties and Selectors on messages that influence the observable behavior of the service
  - Connectors to link web service operation interactions (eg producer to consumer)
  - Dynamic participation
  - Operational context

WSCI example





Ref http:// http://www.w3.org/TR/wsci/

WSCI example

interface contains processes that describe the dynamic behavior

Process models the observable behavior

Instantiation triggers process

Actions executed sequentially

Travel Agent executes the *OrderTrip* operation

BookSeats process is called as part of this action

correlate Confirmation with Order itinerary

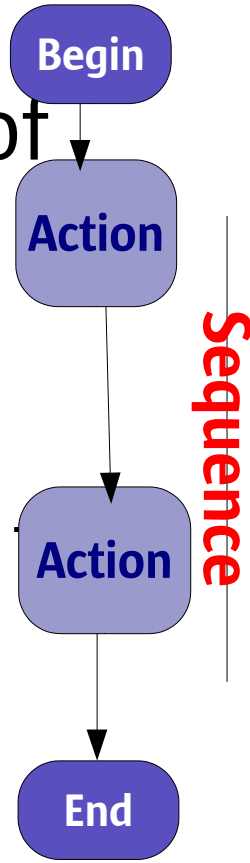
associates action with WSDL operation

```
<interface name="TravelAgent">
   <process name="PlanAndBookTrip" instantiation="message">
      <sequence>
         <action name="ReceiveTripOrder"
              role="tns:TravelAgent" operation="tns:TAtoTraveler/OrderTrip">
         </action>
         <action name="ReceiveConfirmation"
              role="tns:TravelAgent" operation="tns:TAtoTraveler/bookTickets">
         <correlate correlation="tns:itineraryCorrelation"/>
         <call process="tns:BookSeats" />
         </action>
         <action name="SendStatement"
              role="tns:TravelAgent" operation="tns:TAtoTraveler/SendStatement"/>
         </action>
      </sequence>
   </process>
   <process name="BookSeats" instantiation="other">
       <action name="bookSeats"
            role="tns:TravelAgent" operation="tns:TAtoAirline/bookSeats">
       </action>
   </process>
</interface>
```

Ref http://www.w3.org/TR/wsci/

XLANG (from Microsoft)

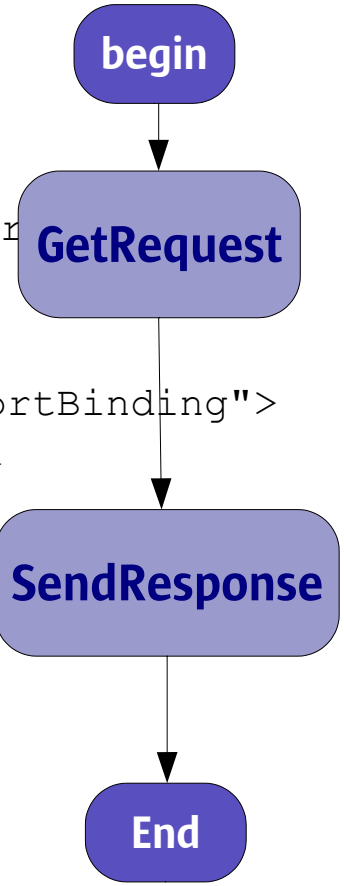
- Xml notation for the specification of message exchange behavior among participating web services in a business process.
- extends WSDL by describing the behavior of a web service.
- Basis of automated BPM:
  - for tracking the state of processes
  - Control of the Actions in the business process to be executed in sequence
  - enforcing correct message flows
- Based on pi-calculus mathematical model



Ref http://www.gotdotnet.com/team/xml_wsspecs/xlang-c

XLANG example

```
<service name="StockQuoteProviderService">
  <port name="pGetRequest" binding="tns:RequestReceivePor
   <soap:address location="mailto:quote@example1.com"/>
  </port>
  <port name="pSendResponse" binding="tns:ResponseSendPortBinding">
     <soap:address location="mailto:response@example2.com"/>
  </port>
  <xlang:behavior>
    <xlang:body>
      <xlang:sequence>
        <xlang:action operation="AskLastTradePrice"
          port="pGetRequest" activation="true"/>
        <xlang:action operation="SendLastTradePrice"
          port="pSendResponse"/>
      </xlang:sequence>
    </xlang:body>
  </xlang:behavior>
</service>
```

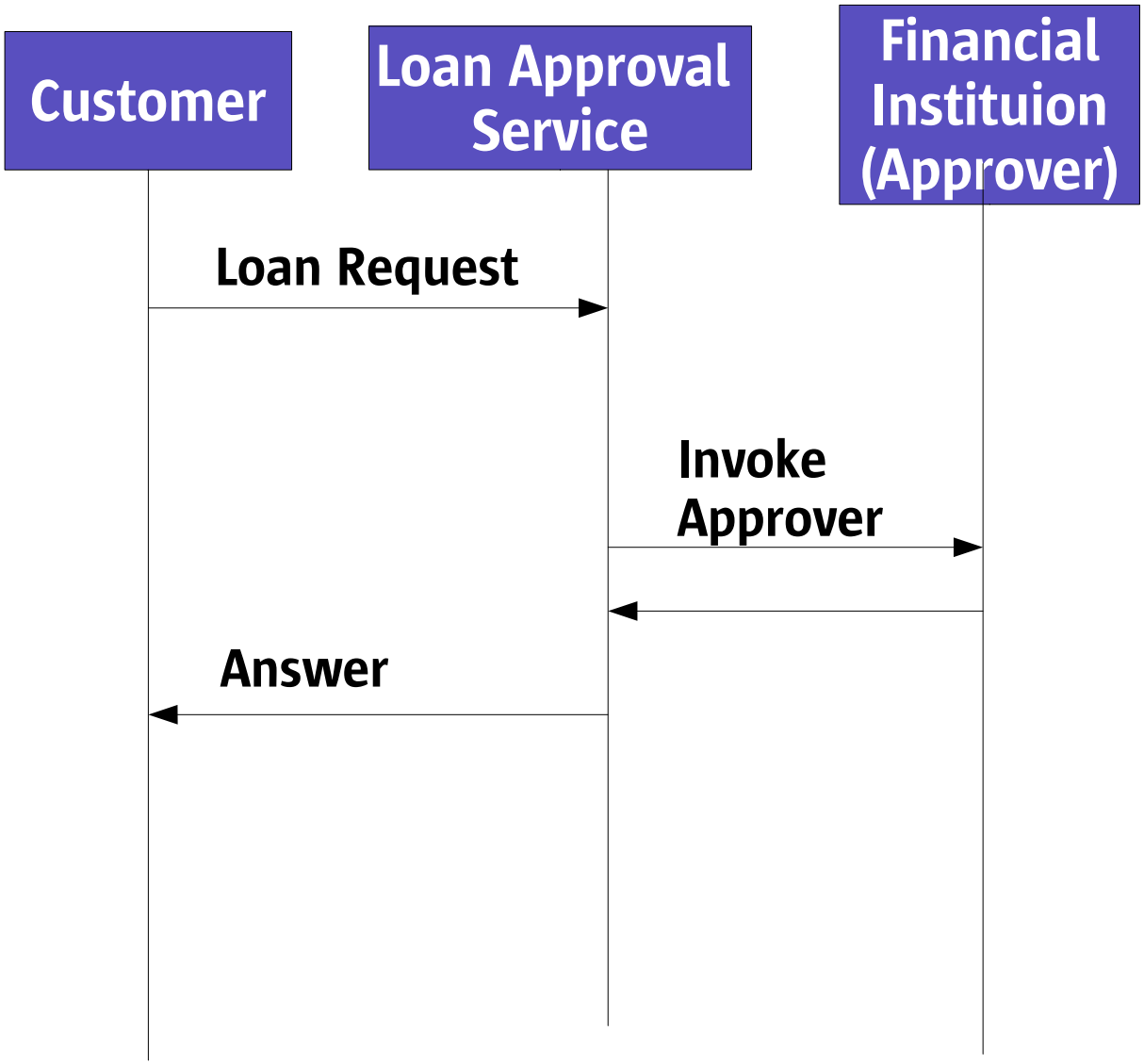


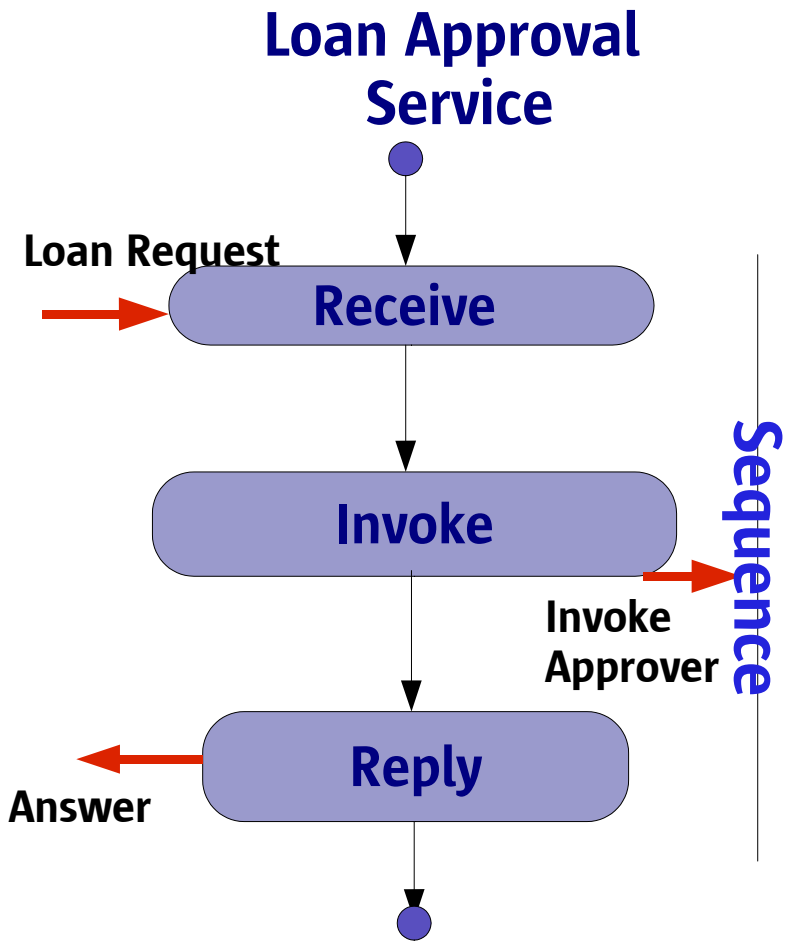
ref: http://www.gotdotnet.com/team/xml_wsspecs/xlang-c

BPEL4WS (from MS, IBM, BEA)

- Combines
  - WSFL (support for graph oriented processes)
  - XLANG (structural constructs for processes)
- allows specifying business processes:
  - execution order of operations of Web services from a collection of Web services
  - the data shared between these Web services
  - which partners are involved and how they are involved
  - joint exception handling

BPELS4WS example





Ref http://www-106.ibm.com/developerworks/webservices/library/ws-bpelcol2/

BPEL4WS example

process include namespaces to refer to required WSDL

Roles refer to portTypes of services Linked

Declares the parties involved

how the partner and the process will interact given the *serviceLinkType*

Data is written to and from containers

receive message from partner at portType

Activities executed sequentially

invoke partner Web service operation at portType

Reply with container data from portType operation to partner

```
<process name="loanApprovalProcess" xmlns="...>
  <partners>
    <partner name="customer" serviceLinkType="lns:loanApproveLinkType" myRole="approver"/>
    <partner name="approver" serviceLinkType="lns:loanApprovalLinkType" partnerRole="approver"/>
  </partners>
  <containers>
    <container name="request" messageType="loandef:CreditInformationMessage"/>
    <container name="approvalInfo"  messageType="apns:approvalMessage"/>
  </containers>
  <sequence>
    <receive name="receive1" partner="customer"  portType="apns:loanApprovalPT"
             operation="approve" container="request" createInstance="yes">
    </receive>
    <invoke name="invokeapprover" partner="approver" portType="apns:loanApprovalPT"
            operation="approve" inputContainer="request" outputContainer="approvalInfo">
    </invoke>
    <reply name="reply" partner="customer" portType="apns:loanApprovalPT"
           operation="approve" container="approvalInfo">
    </reply>
  </sequence>
</process>
```

What is B2B Collaboration?

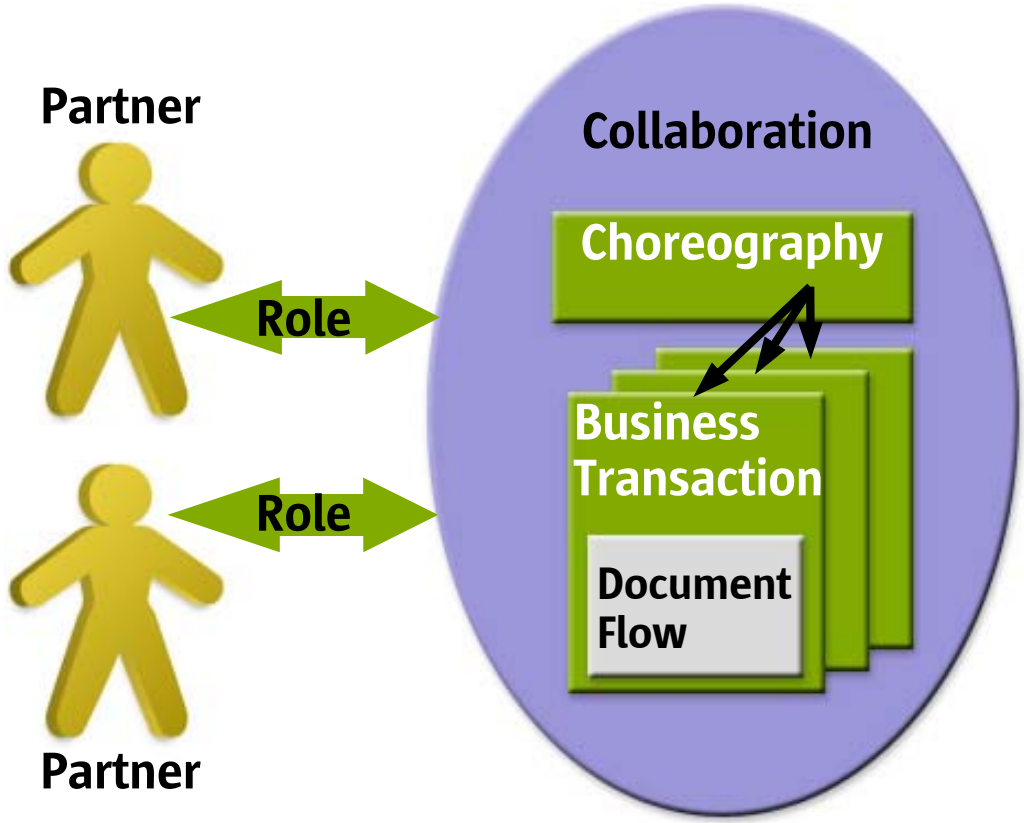
What is Collaboration?

- The agreement among a set of participants (eg Web services) to achieve a common goal or specified outcome in a shared process

What is Collaboration?

Business Process collaboration:

The sequence in which
Messages are exchanged
between roles to support
a business process:



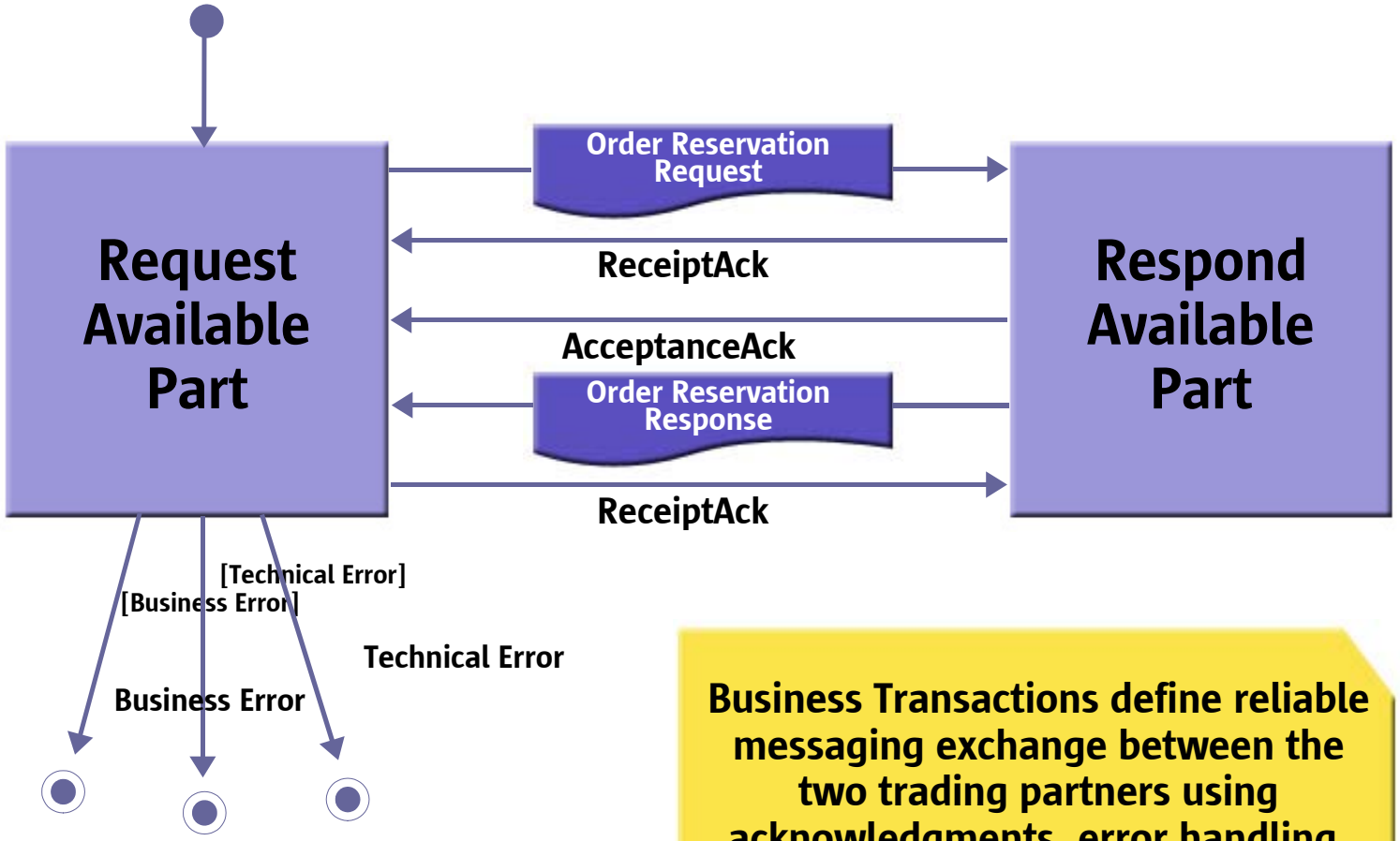
What is BPSS?

EbXML: Business Process Specification Schema

- BPSS defines the collaborative process :
  - Defined in terms of:
    - Sequence of Business Transactions (Message exchanges)
    - Message types
    - Message contents

Business Transaction Definition

comprised of a message request and one or more responses

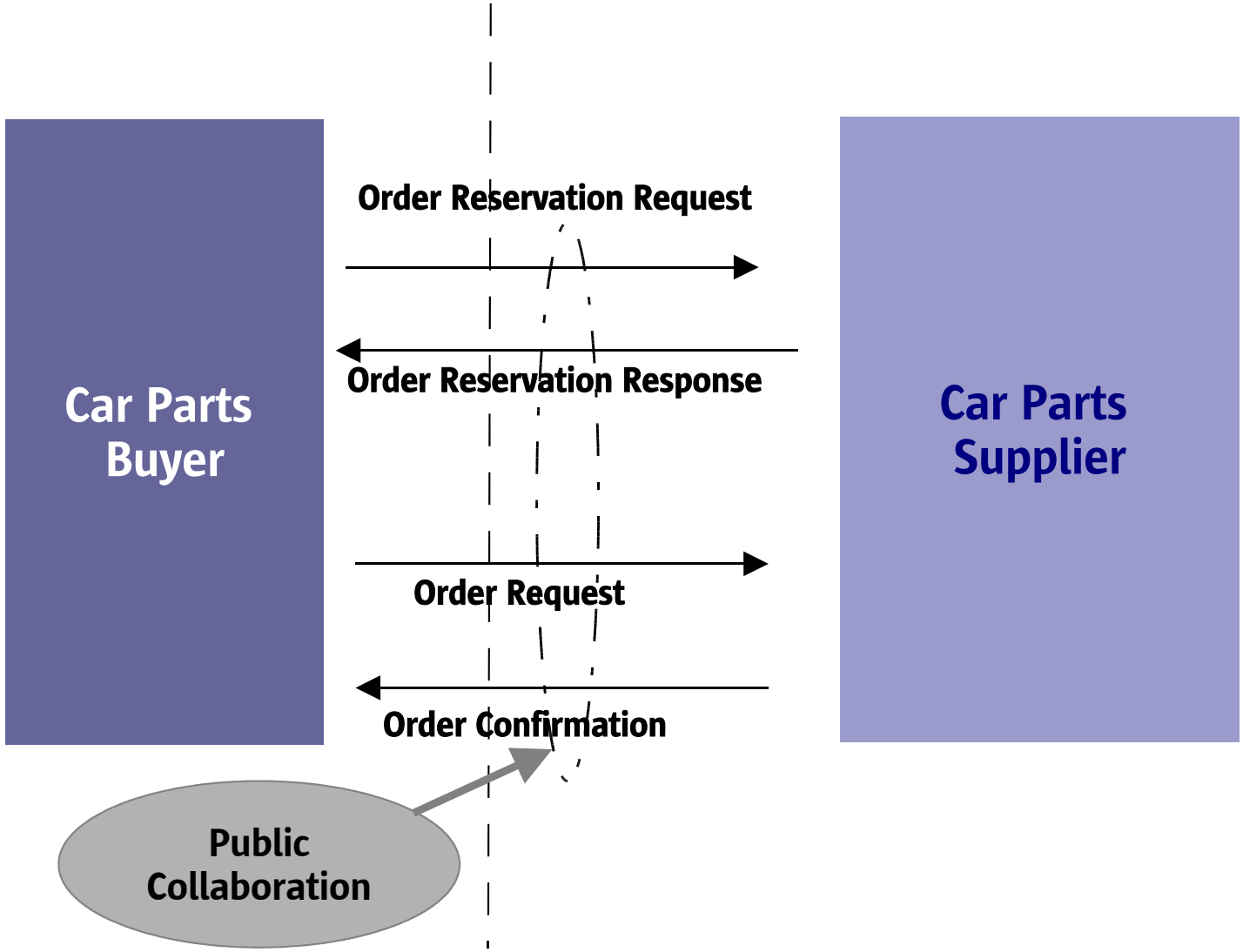


Business Transactions define reliable messaging exchange between the two trading partners using acknowledgments, error handling, logging and roles

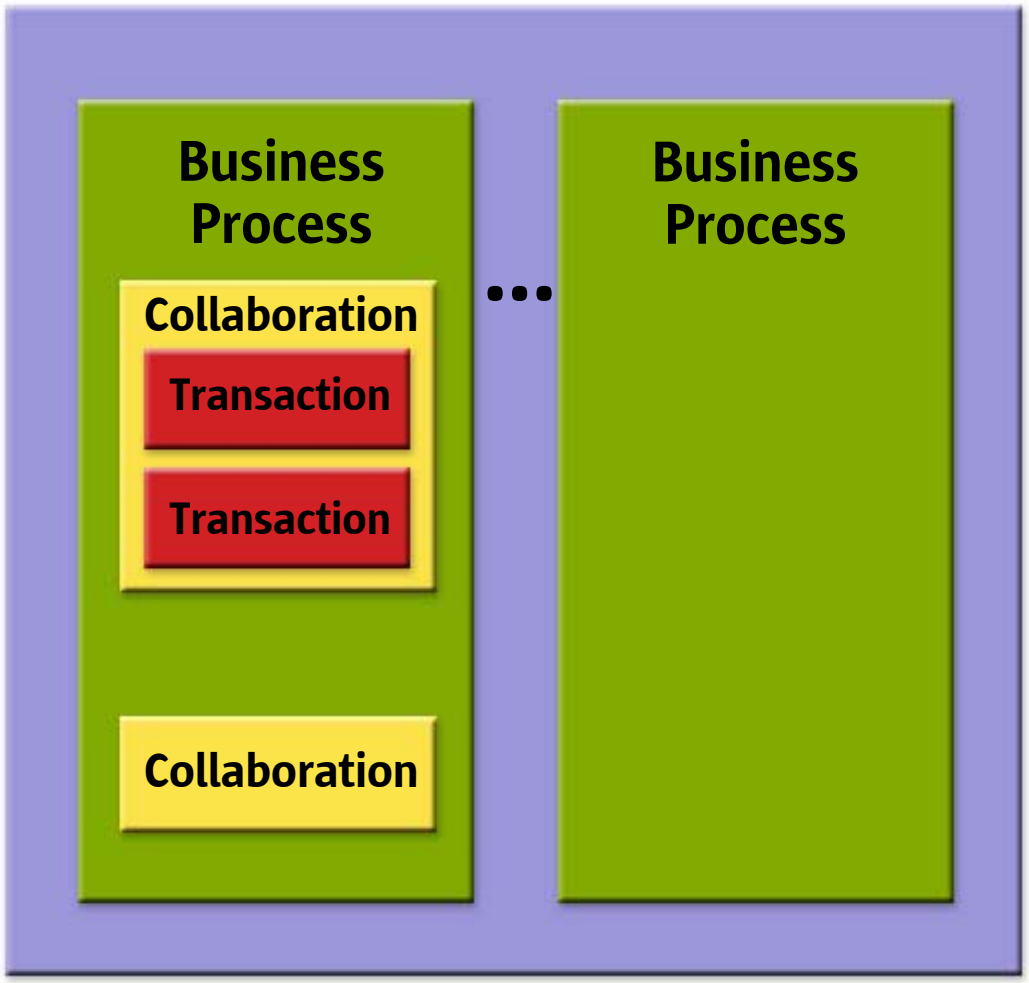
Collaboration Example

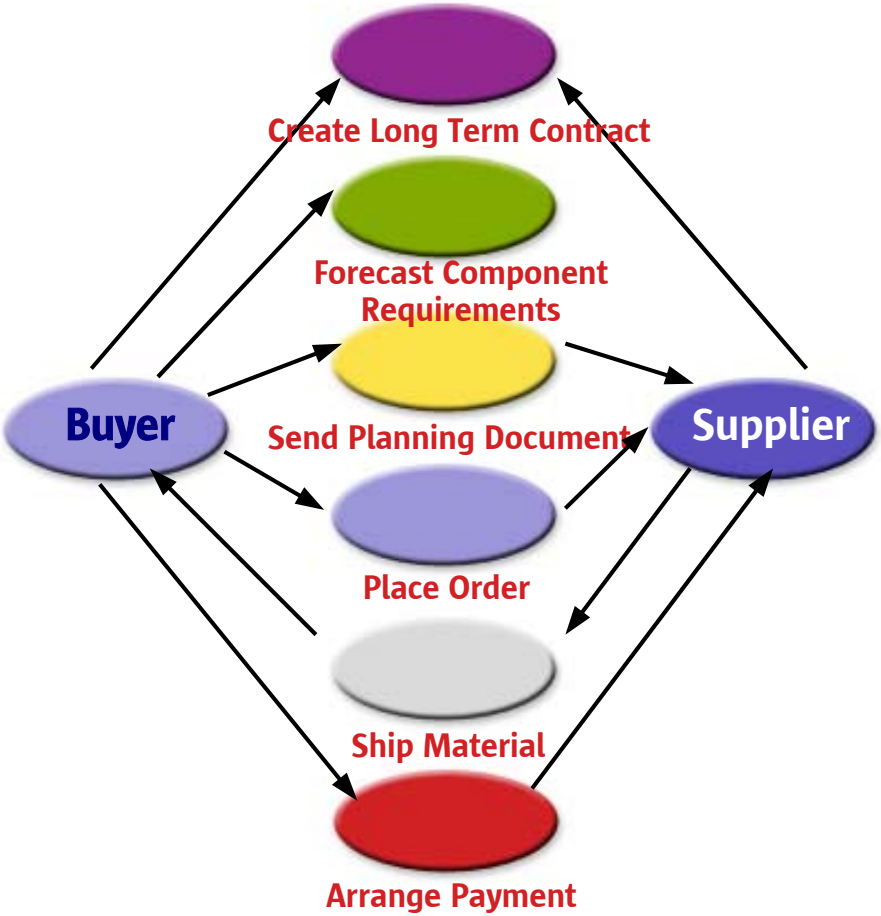
The B2B Collaboration may be composed of several business transactions

The resulting sequence is captured in a BPSS



Business Process Specification



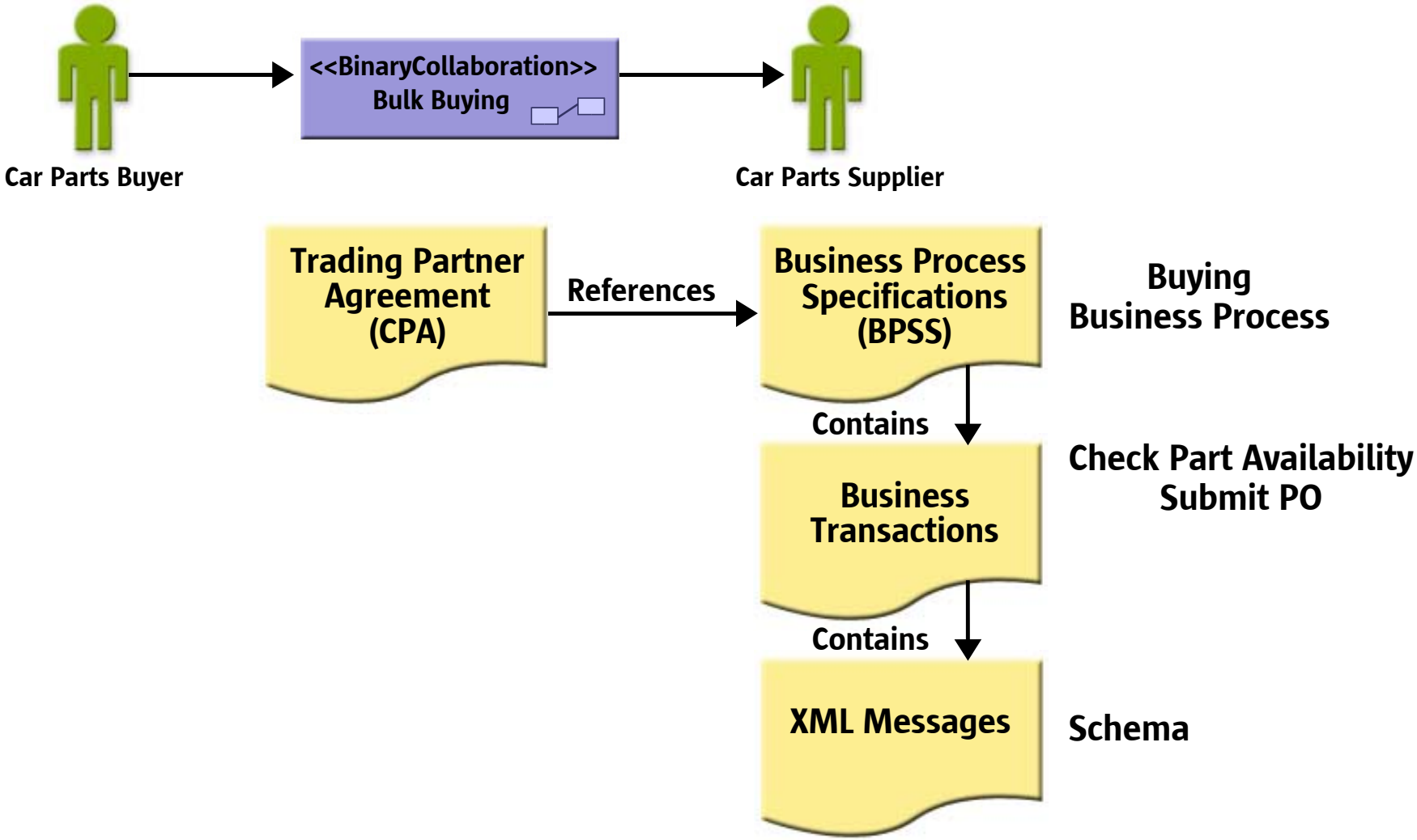


How BPSS? Example

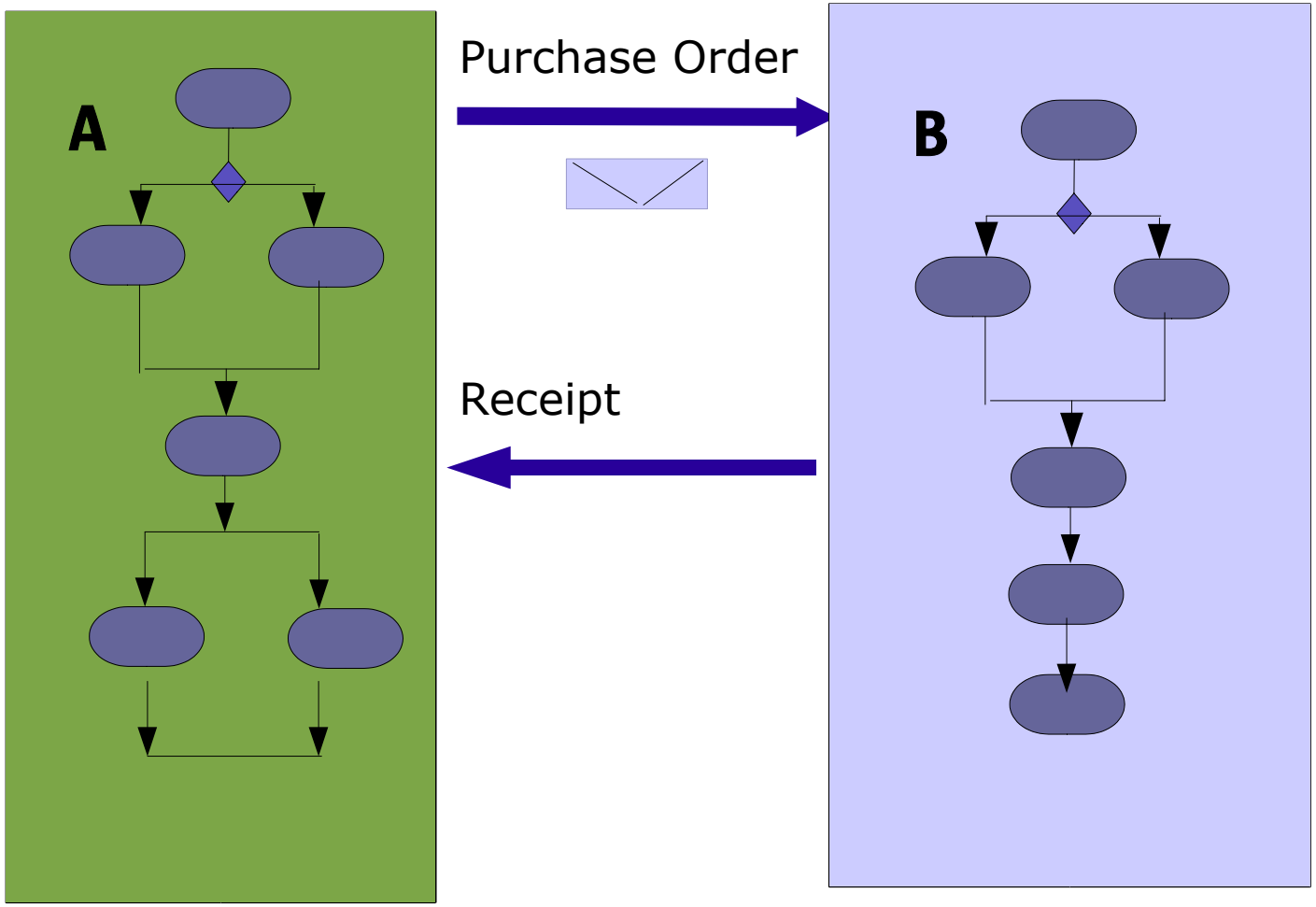
Business Process Specification Example XML (Partial)

```
<BusinessTransaction name="Create Order">
    <RequestingBusinessActivity name=""
            isNonRepudiationRequired="true"
            timeToAcknowledgeReceipt=”P2D"
            timeToAcknowledgeAcceptance=”P3D">
        <DocumentEnvelope  BusinessDocument="Purchase Order"/>
    </RequestingBusinessActivity>
    <RespondingBusinessActivity name=""
            isNonRepudiationRequired="true"
            timeToAcknowledgeReceipt=”P5D">
        <DocumentEnvelope isPositiveResponse="true"
            BusinessDocument="PO Acknowledgement"/>
        </DocumentEnvelope>
    </RespondingBusinessActivity>
</BusinessTransaction>
```

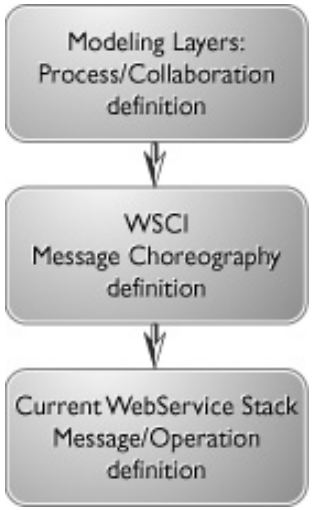
Trading Partner Agreement

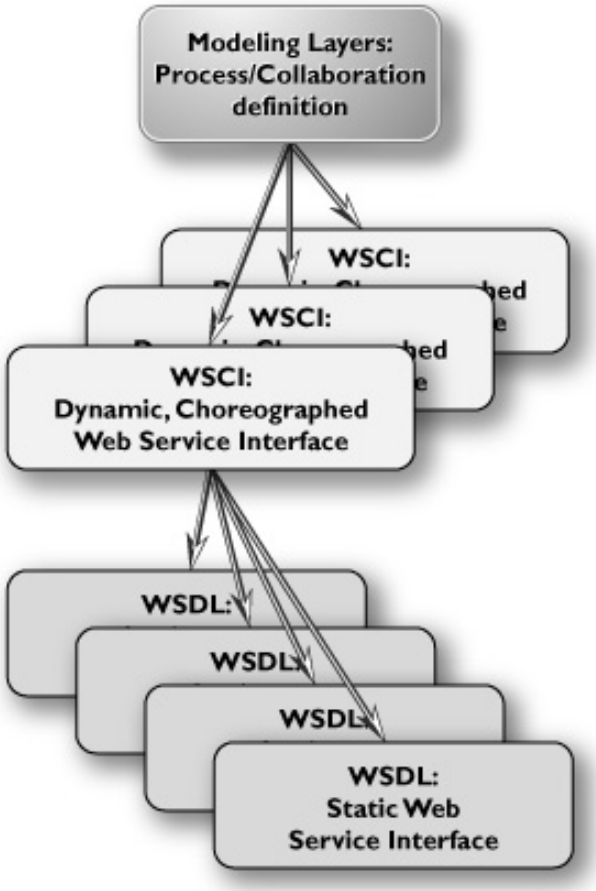


Mapping of Collaboration Messages to Internal Business Processes



Collaboration, Choreography, and Orchestration





BPSS defines semantics and interoperability for business processes and collaborations in a top-down approach. Orchestration/Choreography is more of a bottom up approach or one business' view of a process. It is anticipated that these two views will meet in the middle. WSCI provides the first step to linking the two.

Conclusion

- Orchestration, choreography and collaboration are different views and methods for the composition of web services into business processes
- Orchestration is ~ internal view of executable business process
- Choreography is ~ one partner's view of external composition of business process
- BPSS is for B2B, with legal agreements
  - Two way view of a business collaboration

Carol McDonald

This box provides space for call to action text, URLs, or any relevant info

Sun microsystems
We make the net work.