Service Composition - Outlook

- [ACKM04] Ch 8 (continued)
- [ACKM04] Ch 9
Composition Models

- Dimensions of Composition Model:
  - Component
    - (Web Service)
  - Orchestration
    - activity diagram, Petri net, pi-calculus, ...
  - Data
    - specification and exchange
Composition Models - II

- Service Selection
  - static or dynamic
- Transactions
  - transactional semantics of composition
- Exception Handling
  - abortion and recovery
Service Selection

- Static binding
  - service at fixed URL
- Dynamic binding by reference
  - service URL in a process variable
- Dynamic binding by lookup
  - look up service URL in directory
- Dynamic operation selection
  - like procedure valued variable
Dynamic Binding by Lookup

Newly added node that accesses a (statically specified) UDDI registry to determine which warehouse should be contacted for the product being ordered. The result is stored into the warehouse process variable. Note that in practice several invocations of the UDDI API may be needed to get the desired information.

Variables:
- warehouse: URI
- inStock, shippingAvail: bool
- customer: String
...

Subsequent nodes can use the reference approach and determine the URI based on the value of the warehouse variable.
Transactions

- ACID transactions
- Sagas (compensation)
ACID Transactions
Sagas

Long-lived transaction $T$ (saga)

Forward execution

Compensation flow
Sagas

- Not highly successful in past
- Web Services trend: service responsible for
  - transaction
  - compensation
- (but note WS-Transaction compensates all-at-once rather than in reverse sequence)
Exceptions

- Flow-based
  - test return values and write code ...
- Try-catch-throw
  - e.g. Java
- Rule-based
  - ECA
Coordinating and Composition

- Already said: composition is private, coordination is public ...

- So: must design composed services that can participate in coordination protocols
Procurement Coordination Protocol

- **customer**
  - requestQuote (to supplier)
  - orderGoods (to supplier)
  - makePayment (to supplier)
  - confirmShipment (to warehouse)

- **supplier**
  - checkShipAvailable (to warehouse)
  - confirmOrder (to customer)
  - cancelOrder (to customer)

- **warehouse**
  - orderShipment (to warehouse)
  - getShipmentDetails (to customer)
  - confirmShipment (to supplier)

- warehouse confirms
- warehouse cancels
Supplier View

- customer
  - requestQuote (to supplier)
  - orderGoods (to supplier)
  - makePayment (to supplier)

- supplier
  - checkShipAvailable (to warehouse)
  - confirmOrder (to customer)
  - cancelOrder (to customer)
  - orderShipment (to warehouse)

- warehouse
  - confirms
  - cancels
  - confirmShipment (to supplier)

- orderGoods (to supplier)
Translate Operations to Activities

- Request/Reply from R -> Receive/Reply
- Send from R -> Receive
- Send to R -> Send
- Request/Reply to R -> Invoke
Result of Translation

- receive requestQuote
- reply requestQuote
- receive orderGoods
- invoke checkShipAvailable
- send cancelOrder
- send confirmOrder
- receive makePayment
- send orderShipment
- receive confirmShipment
From Protocol to Executable Process

receive requestQuote
invoke lookupQuote
reply requestQuote
receive orderGoods
invoke checkShipAvailable

shippingAvail=true
send confirmOrder
receive makePayment
invoke collectPayment
send orderShipment
receive confirmShipment

shippingAvail=false
send cancelOrder
Where is Conversation Control?

- Unfortunately, two places ...
(Recall) WS Conversation Controller

**service provider**

- object (Web service implementation)
- horizontal protocol implementation
- conversation routing, compliance verification

B: conversation compliant with a business protocol
H: conversation compliant with an horizontal protocol
(Recall) Composition Middleware:

- Service composition model and language (usually characterized by a graphical and a textual representation)
- The run-time environment executes the Web service business logic by invoking other services (through SOAP and HTTP modules)

Web service composition middleware:
- Development environment
- Run-time environment (composition engine)
- Schema definitions
- Composite service execution data

Other Web services middleware (e.g., SOAP engine and conversation controller)

Services offered by other providers:
- Supplier
- Warehouse
- Accounting

A service provider

Schema designer
Put Them Together ...

• Composition Engine is required to do routing without an explicit context ID ...
• e.g. Application’s orderID
Better ...

service provider

object (Web service implementation)

Instance of a composition schema

B, H

composition engine

conversaion controller

B, H

B

H

service requestor

B: conversation compliant with a business protocol

H: conversation compliant with a horizontal protocol
- **Business Process Execution Language**
- **WS Specification du jour for Composition**
Scope of BPEL

Abstract and/or executable process orchestration, variables and data transfers, exception handling, correlation information (for instance routing)

Roles:
- customer
- warehouse
- local service offered by supplier

Port types:
- receive orderGoods
- invoke checkLocalStock
- invoke checkShipAvailable
- invoke cancelOrder
- invoke confirmOrder

Variables:
- warehouse: URI
- inStock, shippingAvail: bool
- customer: String
...
BPEL Orchestration Activities

• Basic Activities
  • invoke / receive / reply
  • assign / wait
• Structured Activities
  • sequence
  • switch / pick
  • flow
  • while (!)
Example

processOrder
  sequence
    receive orderGoods
    invoke checkLocalStock
  
chooseLocal
  switch
    inStock=false
      searchExternal
        sequence
          invoke checkShipAvailable
    inStock=true
      invoke confirmOrder

chooseExternal
  switch
    shippingAvail=true
      invoke confirmOrder
    shippingAvail=false
      invoke cancelOrder
BPEL Data Transfer

- Blackboard approach
- Data types are WSDL message types
- Assignment
  - input / output parameters
  - explicit using assign activities

<table>
<thead>
<tr>
<th>custName</th>
<th>“Homer Smith”</th>
</tr>
</thead>
<tbody>
<tr>
<td>partNum</td>
<td>123-456-98765</td>
</tr>
<tr>
<td>quantity</td>
<td>3</td>
</tr>
<tr>
<td>etc</td>
<td></td>
</tr>
</tbody>
</table>

Values have WSDL message types
BPEL Service Selection

- Partner Link - type and definition

**partner link definition**: Further qualifies the interactions occurring through a partner link type. Its definition refers to a partner link type and specifies the role played by the composite service as well as the one played by the other partner.

```
<partnerLink name="customerP"
  partnerLinkType="orderLT"
  myRole="supplier"
  partnerRole="customer">
</partner>
```
BPEL Scopes, Exceptions, Transactions

- Nested Scopes
- Exception Handling
- Transaction
- Compensation
Scopes Example - Exceptions

receive orderGoods
invoke checkLocalStock
invoke checkShipAvailable

chooseLocal
switch
inStock=false
searchExternal
invoke confirmOrder
chooseExternal
switch
inStock=true
invoke confirmOrder

shippingAvail=true
invokes confirmOrder
shippingAvail=false
invoke cancelOrder

scope of the searchExternal activity

due to the behavior of the default handler, implicitly associated
with each activity, a fault F occurring in activity send confirmOrder would
propagate up until activity searchExternal, where the handler resides.
Routing

- Define Correlation Sets
- set of data items used as routing key

- orderID can be used for correlating the two messages

```plaintext
message checkAvailability
orderID
requestedDeliveryDate
deliveryLocation
...
```

```plaintext
message availability
orderID
shippingAvail
```
Outlook

• Chapter 9 ...
What is available now?

- SOAP
- WSDL
- UDDI
  - still mainly design-time discovery
- WS-Coordination, WS-Transaction, ...
  - backfilling, adding P2P
- BPEL
  - promising
Current Situation ...

- SOAP/WSDL/UDDI as currently used ...
  - just a translation of traditional 2-tier client-server RPC
  - more widely-adopted standard than its predecessors
The Easy Cases

- EAI for legacy client-server applications
- Exposing client-server applications on the Web
- Both are easy extensions of existing middleware
The “Holy Grail”

- Dynamic service discovery and invocation
- In practice we are still very far from this goal!
B2B - Ontology

• In XML
  • is it “instructor” or “professor”?
  • note these are equivalent in some contexts and different in others

• In WSDL message type
  • is “price” in dollars or Euros?
  • does it include VAT?

• This is a huge issue!
  • will require a “revolution” in technology
Avoiding the Problem

- Small closed communities
- Agreement is manageable
- One dominant entity
- All agree with the 800 pound gorilla
Addressing the Problem

- Middlemen
  - rating services, market makers, hubs
- Many unsolved issues of trust / confidentiality
Addressing the Problem - Semantic WS

- DAML+OIL
  - DARPA Agent Markup Language
  - Ontology Interface Layer
- OWL
  - Ontology Web Language
  - Complexity => 3 sublanguages, not all decidable!
- Etcetera
  - this is still far from real
Management

• Important issue in complex distributed systems
• Traditional: SNMP, CIM
  • instrumentation and control interfaces
• What about WS?
Local WS Management Architecture

- where confidentiality and trust not at issue
Cross Enterprise Management

• Must support
  • Measurements (e.g. load)
  • Events (e.g. service abort)
  • Policy (< 6 logon attempts)
  • SLA (desired QoS guarantees)

• Must address confidentiality and trust
Trusted Third Party

- These protocols not yet defined
Management Protocols are WS

- (just an obvious remark that does not really address any of the difficult issues)
Another Approach

- Tunnel data from legacy management protocols (CIM) between managers and data repositories (XMLCIM)

Management systems

XMLCIM (XML over HTTP)

CIM data model repository

Proprietary protocols to populate repository

Managed resources
• Open Grid Services Infrastructure
  • originally shared computing cycles
  • now considerably more ambitious
• Resource Management
• Extensions to WSDL
  • attributes attached to port types for events, data collection
• Required Port Types
  • clients must interact with a service using resource manager
The End

• of [ACKM04]