Web Services: Life Cycle Intelligence

A/Prof. Boualem Benatallah

School of Computer Science and Engineering
The University of New South Wales (UNSW)
Agenda

- Service Oriented Computing: principles and issues
- ServiceMosaic Project: Service Lifecycle Intelligence
Material based on work in ServiceMosaic project, collaboration with:

- Dr. Fabio Casati (HP Labs, USA)
- Dr. Regis Saint-Paul (UNSW, Australia)
- Mr. Hamid Motahari (UNSW, Australia)
- Ms. Woralak Kongdenfha (UNSW, Australia)
- Mr. Julien Ponge (LIMOS, France)
- Dr. Farouk Toumani (LIMOS, France)
- Prof. Mohand-said Hacid (LIRIS, France)

Web sites
www.cse.unsw.edu.au/~soc
servicemosaic.isima.fr
Development of Integrated Applications

• Applications and data sources are autonomously developed and deployed
• Proprietary technologies (communication protocols, data formats)
• Rigid infrastructures, costly development and maintenance of integrated systems
Service Oriented Architectures (SOA)

- Standard protocols and languages
- Agile infrastructures
Principles

• *Facilitate programmatic interactions of autonomous systems* (e.g., data-intensive, computational resources) via resource virtualization, infrastructure simplification and consolidation

• Web services: *self-described* software entities (at least best possible description), that can be published, discovered, and invoked over the network (using XML-based standard languages and protocols)

• On demand computing through composition

• *Services are poised to be as ubiquitous as the Internet* (e.g., payroll, on-board systems, e-government, services at your computer)
Issues Relevant to Various Service Life Cycle Activities

Semantic Interoperability

SOA solved low level interoperability but for programming by composition to scale, services should be used with assistance from providers.

Richer component model (possible interactions, OoS, data types)
Various integration layers (e.g., data, process)

Aligning IT services with business services

Correlation of higher level abstractions to lower-level system observable events.
Continuous monitoring of resources and processes

Evolution and change management

Contract violation, impact analysis and process migration
Current Standards Maze

Too many, Overlap, Conflicts, Only few standards are used (WSDL, SOAP)
(See "Interoperability Specifications", May 06, IEEE Computer, by Motahari, Benatallah, Casati and Toumani)
Service Oriented Technologies - Layers

Discovery
- UDDI
- ebXML registry

Contracts
- BPEL4WS
- BPML
- OWL-S service model
- BPML

Business process
- WS-Transaction
- BTP
- OWL-S service model
- BTP

Transactions
- WS-Coordination
- WSCI
- OWL-S service profile
- ebXML BPSS

Choreography
- CS-WS
- WSCL
- OWL-S service profile
- ebXML messaging

Conversations
- WSEL
- WSEL
- OWL-S service grounding
- ebXML CPP

Nonfunctional description

Service description

XML-based messaging

Network
- SOAP
- HTTP, FTP, SMTP, and others

Source (Turning Software to a Service, M. Turner et al., IEEE Computer, Vol 36(10), 03)
Systems Integration: B2B Integration Example

Private process (Company-specific)

- Process
  - Receive PO Request
  - Select Supplier
  - Generate RFQ
  - Send RFQ
  - Select RFQ Response
  - Send PO
  - Close

Public process (Standard)

- Send PO Customer
  - Send PO
  - Receive PO
  - Acknowledge
  - Receive PO Response
  - Send PO Response
  - Send PO Response Acknowledge

Public process (Standard)

- Send PO Supplier
  - Receive PO
  - Acknowledge
  - Send PO Response
  - Receive PO Response
  - Send PO Response Acknowledge

Private process (Company-specific)

- Process Sales Order
  - Receive PO
  - Check Customer
  - Check Credit
  - Check Availability
  - Create Sales Order
  - Send PO Response
  - Close

(Source: e-business Architectures and Standards, Anil L. Nori, Tutorial, VLDB’2002, HongKong, China)
Integration Layers

Business Partner 1

PeopleSoft

Internal system

Workflow

Business rules

Programs

External interactions

Content of document

Message definition

Business Protocol

Policy

External interactions

Content of document

Message definition

Business Protocol

Policy

Internal system

Workflow

Business rules

Programs

Database

Adapter

Legacy applications

Business Partner 2

SAP/R3

Adapter

©Boualem Benatallah, 2006
Business Protocol Layer (Cont.)

- **invoke**
- **receive**
- **invoke**
- **receive**
- **receive**
- **invoke**
Some Observations on Integration Technologies

- Great deal at transport protocol level, process-based orchestration, programing support for dynamic port binding
- While standardisation makes interoperability easier, it does not remove the need for data transformation, protocol adaptation
- Services are designed to be open (possibly no precise knowledge of the various data models and protocols of clients)
- Data is organized by application, not by following agreed upon conceptual models
- Standardization of protocols at lower level and languages for higher level abstractions
ServiceMosaic Project: Services LifeCycle Intelligence

Modeling of business protocols
Algebra for protocol models management
Model-driven and aspect-oriented approach to support service adaptation
Model-driven change impacts analysis
Protocol discovery
Query and visualization support for service monitoring, property discovery and verification
Conversations and business protocols

- protocols describe valid ordering of messages
  - stateful interaction
  - this must be done somehow, formally (best) or informally
  - coarse-grained
- complements interface definition and other specs (semantics)
Protocol languages

invoke requestQuote
reply requestQuote
receive orderGoods
invoke checkShipAvailable
invoke cancelOrder
invoke confirmOrder
receive makePayment
invoke orderShipment
receive confirmShipment
Protocols and choreographies

A: Protocol that a service obeys to when interacting with a client

B: Protocol that a service obeys to when interacting with its clients

C: a choreography

D: The protocol that a Business service obeys to when interacting with one or more clients

“Business” service
Business Protocols: benefits of explicit representation

• And they are quite a lot!

- service development, management, evolution, binding
- Protocol-aware tools can make life easier for developers
Development support

top down

Protocol specs

protocol compiler

service skeleton

add biz logic

service implementation

bottom up

Protocol specs

protocol generator

service implementation

© Boualem Benatallah, 2006
Generating composition skeletons

customer

requestQuote
orderGoods
confirmOrder or cancelOrder

supplier

receive requestQuote
reply requestQuote
receive orderGoods

send cancelOrder
send confirmOrder
Avoid implementing exception handling logic
Static and dynamic binding
Versioning, evolution, compliance

customer

version 1.0

version 1.1

XYZ standard specs

© Boualem Benatallah, 2006
Expressive power

- Services are designed independently of clients
- Need to be (fairly) fully specified
- Ordering is not enough.
- Implicit (timed) activations, timeouts
- Message content-based conditions, state management
  - hiding details (multiple views)
- Transactional implications
- Rightsizing is the key
Business Protocols Modeling: Embryonic Model

- **Start**
  - T1: Login
  - T3: AddToCart()

- **Search book**
  - T2: SearchBook()

- **Book selection**
  - T4: SearchBook()
  - T5: RemovefromCart()
  - T6: OrderBook()
  - T8: CancelBookPurchase()

- **Ordered**
  - T8: CancelBookPurchase()
  - T7

- **Cancelled**
  - T8: CancelBookPurchase()

- **Shipped**
  - T9: ReturnBook()

- **Returned**
  - T9: ReturnBook()

- **Completed**
  - T10
Protocol models management

- Bring the benefits of abstractions to service-oriented systems developers
- Analogy with data management: simple model (relational), declarative language (SQL) and algebra
- Operators to manipulate protocol models (e.g., identifying similarities and differences, identify patterns, identify relationships)
- Semantics and operators defined based on labeled transition systems
Protocol models Management (cont.)
Summary

- Service oriented computing offers tremendous opportunities for agile and Internet scale integration
- Business process modeling, web services offer key solutions to virtualization and high level orchestration of business processes
- Business activity management techniques offer key solutions to business process monitoring and management
- Effective automation requires an integrated framework that offers lifecycle intelligence! : conceptual modeling, algebra for analysis and manipulation of process views, discovery and debugging, model-driven development of adaptors, simulation, service level-operational level mappings, UI models management and context support for services, ...