The Role of BPM in Service Oriented Architecture

Charlton Barreto

Adobe Senior Computer Scientist/Architect

charltonb@adobe.com

Agenda

- Introduction
- Convergence of BPM and SOA
- Modeling
- Composition and refinement
- Service constraints/semantics
- Wiring applications together
- Transaction models

Why SOA?

- SOA = an architectural principle for structuring systems
- SOA emphasizes the de-coupling of system components
- New services are created from existing ones in a synergistic fashion
- Strong service definitions are critical
- Services can be subsequently re-composed in response to changing business requirements

Why composite applications?

- Composite applications are...applications!
  - Build on loosely coupled services
  - Service sets – Services of services ☺
- Comprised of heterogeneous parts
  - Existing services
  - New services
- Composite applications != SOA
- Composite applications employ SOA principles
  - Features exposed as Web services
  - Standards-based interaction between services
  - Described by standards-based artifacts
  - Are themselves composable

Why BPM?

- BPM describes how systems perform their business processes
- SOAs require understanding of roles, business expectations, interoperability
  - BPM serves these functions in a SOA
- With BPM, it is straightforward to let your business partners and customers directly participate in your business processes
- For BPM, services can be associated with business processes, which act on the behalf of activities

Business-level modeling

- Business-level modeling comes with a set of methodologies
- Different methodologies compete
  - BPBM
  - BPMN
  - Semantic Web
  - UMM
- Large number of business-level methodologies
  - Chances to unify them is very low
- Very likely, there will always be different business-level modeling tools

Business processes need to

- Co-ordinate asynchronous communication between services
- Correlate message exchanges between parties
- Implement parallel processing of activities
- Implement compensation logic (“Undo” operations)
- Manipulate/transform data between partner interactions
- Support long running business transactions and activities
- Provide exception handling
- Provide universal data model for message exchange

Roles of modeling

- Enterprise Information Systems (EIS) are developed to automate or support business processes
  - Need to be modeled, verified and validated before
- EIS manages enactment
  - Model is the configuration parameter
- Components of an EIS cooperate by means of middleware
  - Models are the configuration parameters of orchestration and choreography

Types of models

- BPMN
- ebBP
- CDL
- BPEL
- BPEL4People
- UML Activity Diagramme
- UML State Chart

BPMN

- Standard Business Process Modeling Notation
- Addresses graphical elements
  - Flow Objects
  - Connecting Objects
  - Swimlanes
  - Artifacts
- No formal interaction model
- Mapping to BPEL 1.1

ebBP

- Basis of eBusiness collaboration
- Defines interoperable business processes through document-based message exchange
  - Standard and extensible business transaction patterns
  - Support for multiple role bindings
  - Flexibility for complex transaction activities
  - Business quality of services *for* services
  - State alignment using business signals and business

CDL

- Peer-to-peer global model operating between loosely coupled components
  - Established global behavioural contract
  - Based on formalised description of external observable behaviour
- Describes collaboration protocols
  - Services act as peers
  - Interactions may be long-lived and stateful

BPEL

- Orchestrating interaction between Web services in a business process
- Recursive aggregation model
  - Aggregation: A set of Web services can be tied into one or more new Web service by means of a business process model
  - Recursive: These new Web services can again be tied into other new Web services

BPEL4People

- Extension of BPEL
- Work toward representing human tasks in a BPEL process – *People activity*
- 5 process-task patterns
  - Inline tasks
    - People activity
    - BPEL process
  - Standalone tasks: External to BPEL process
    - People activity - direct
    - People activity - WSDL portType
    - Invoke activity - WSDL portType

Interaction patterns

- Document common problems and approaches related to the design and implementation of processes and services
- Special emphasis on situations that arise when processes and services engage in concurrent and interrelated interactions

Interaction protocol

- Processing behaviour
- Interaction flow
- Interaction role
- Interaction bindings
- Interactive parts – i.e. activities
- Interactive part bindings

Model - Verification

- Services
- Activity & sequence diagrammes
- Interaction protocol

Validation - Correctness

- *Proper completion*: services should always be able to terminate with an end event, and they should not leave any unfinished work
- *Invariants*: constraints on case and base objects should be valid as soon as a service has finished
- Formalisms
  - Pi-calculus
  - Petri nets
- Business-level
  - BPDM Meta model (OMG)
  - UMM Meta model (UN/CEFACT)

Conversation = Series of Related Messages

- Can span seconds, hours, days
- Multiple concurrent conversation instances
- Messages belonging to one conversation are correlated

Describing Conversations

- Message Exchange Patterns
- Business Transaction Patterns
- Collaboration
- Choreography
- Orchestration
- Temporal Logic

Observable behaviour definition

- Interleaves with conversation patterns
- Definitions
  - ebBP
  - CDL
  - BPEL Abstract Processes
  - UML 2.0 Sequence Diagrammes

Conversation Patterns

- Message Exchange Patterns
- Dynamic Discovery
- Renewing Interest
- Reaching Consensus
- Business Exception Handling
- Fault/Signal Handling

Business process complexity

- Complexity
  - Activity
  - Control-flow
  - Data-flow
  - Resource
  - Event
- Metrics:
  Several fields apply
  - Information complexity
  - Cyclomatic complexity
  - Kolmogorov complexity
  - Cognitive complexity
  - Computational complexity

Service constraints/semantics

- Service component architecture
  - Assembly of individual services into larger applications
  - Defining format for configuring a module for deployment
  - Components that are composed into modules can be
    - Custom programs
    - BPEL processes
    - WS-A endpoints
    - Anything else conforming to the interface specification
- *SCA* and *JBI* are applications

Service Bus

- Dynamic connection, mediation and control of services and interactions,
- Spans services via a consistent model
- Enables changing processes and relationships between applications via configuration
- Provides management and monitoring facilities without requiring instrumentation
- *OpenESB* and Apache *ServiceMix* are applications

Transactions – Service-level

- Web Services Transaction 1.1 (WS-TX)
  - Three specifications
    - Web Services Coordination
    - Web Services AtomicTransaction
    - Web Services BusinessActivity
- Specifics of transaction and compensation management is outside the scope of WS-TX

Transactions – Business-level

- ebBP
  - Business transaction protocols
- CDL
  - Choreographies
- BPEL
  - Compensation

Role of BPM in SOA

- BPM provides abstraction for defining business processes
- Services provide the functions to support business processes
- SOA provides capabilities for combining services in an agile, flexible manner