Distributed Systems

Service-oriented Architectures and Business Processes

Tenshi Hara
tenshi.hara@tu-dresden.de
https://www.rn.inf.tu-dresden.de/hara
Outline

- **Services**
  - Service Definition
  - Service-oriented Architectures

- **Business Processes and Workflows**
  - Definitions
  - Business Process Modelling with Petri-Nets
  - BPMN (Business Process Modelling Notation)
  - BPEL (Business Process Execution Language)

- **WS* Specifications**
  - Transaction support
  - Security specifications

Further details on Web Services can be taken from the lecture series ‘Service and Cloud Computing’ offered by Iris Braun.
The Problem with Components

- mainly use of homogeneous component model and platform
- usually bound to one programming language
- close coupling of components within intranets of companies
  - limited flexibility for changing business processes
  - issues with firewalls
  - issues with globally integrating business of different companies
  - issues with heterogeneous enterprise infrastructures
‘A service is a mechanism to enable access to one or more capabilities, where the access is provided using a prescribed interface and is exercised consistent with constraints and policies as specified by the service description.’

- Enabling of access to capabilities
- Service description defining constraints and policies
- Functionality provided by formally specified interface
  - Encapsulated functionality
  - Implementation transparency
- loose coupling
Service-oriented Architectures

- Payment Processing
- Service Interface
- Bank

Business A
- Service Interface
- Order Processing
- Service Interface
- Customer Administration
- Service Interface
- Discount Administration

Warehouse Management
- Service Interface

Supplier
- Service Interface
- Order Processing
Service-oriented Architectures

- Focus on business processes
- Main functionality of applications provided by services
- Services have higher granularity than objects and components
- Encapsulate part of business process functionality
- Loose coupling
  - Services are offered, searched for and used
  - Services may be arbitrarily distributed
  - Services are dynamically composed to business processes
  - Interoperability over platform and business borders
- Business Process: is a ‘collection of activities performed by human users or software applications that together constitute the different steps to be completed to achieve a particular business objective.’ [Alonso et. al]

- Workflow: ‘The terms workflow, workflow process, or sometimes simply process refer to a formal, executable description of a business process.’ [Alonso et. al]
Business Process Management

- Decomposition of business process into sub-processes
- Mapping to service infrastructure required to be executable
- Creation of new services to provide application logic required for business process
- Modeling of the business process

Application Logic

- Web forms
- CRM
- Order Processing
- Payment Processing
- Shipping and Invoicing
- Warehouse Management

Order Handling

- Purchasing
- Customer Management
- Order Processing
- Payment and Invoicing
- Shipping
Business Process Management

Order Handling
- Purchasing
- Customer Management
- Order Processing
- Payment and Invoicing
- Shipping

Business Process Modeling
1. Order verification
2. Risk analysis
3. Calculate discount
- Payment
- Invoicing
- Shipping

Application Logic
- Web forms
- CRM
- Order Processing
- Payment Processing
- Shipping and Invoicing
- Warehouse Management

Business process perspective
Application logic perspective
Classification of services

1. Orchestration Services: Representation of Business Processes

2. Task-centred Business Services: Encapsulation of Business Logic which is not part of Business Process

3. Entity-centred Business Service: Encapsulation of Business Entities (e.g., customer, order list)

4. Application Service: Encapsulation of Application Logic
Business process modeling

- Formal notion based on Petri-Nets
- Basic concepts:
  - **Places** represent states of workflow execution (e.g., results, documents or data buffers)
  - **Transitions** model activities of a workflow
    - Representation of task or basic services to process a workflow
    - Conditions can define input predicates for transitions
  - **Arcs** (Links) connect places and transitions
  - **Markers** model the flow at a certain point in time
    - Places can have markers
    - *Capacity* defines max. number of markers supported in a place
Petri-Net example – Product order

Properties:
- limited (Marker Quantity, here ≤2)
- non-alive (terminating)
- secure (Marker Quantity of each place/state is 0 or 1)
- non-conservative (Marker Quantity varies)
Languages for Business Process Modelling

- Petri-net based modelling languages
- Activity-diagrams in UML → limitations for the modelling of complex processes
- Yet Another Workflow Language (YAWL) → extended petri-nets
- Business Process Modelling Notation (BPMN) and Business Process Execution Language (BPEL)
  - based on flow diagrams
  - Mapping from BPMN to BPEL
  - widespread adoption in industry
Modeling of activities which are performed at certain points in time

Basic Elements
- Flow elements: Event, Activity, Gateway
- Connection Objects: Sequence Flow, Message Flow, Association
- Swimlanes: Pools and Lanes
- Artefacts: Data objects, Group, Annotation

Further elements for expressing timeouts, error handling and transactions
BPMN Example

- **Process Payment**
- **Verify Order**
- **Manage Customer**
- **Calculate Discount**
- **Handle Payment**
- **Arrange Shipping**

**Gateways:**
- **Exclusive Gateway**
- **Parallel Gateway**

**Activities:**
- **Purchasing**
- **Manage Invoicing**
- **Start Event** (type: Message)

**Flows:**
- **Message Flow**

**Event:**
- **End Event**

**Process Elements:**
- **Subprocess** (decomposable)
- **Pool and Lane**

**Diagram:**
- **BPMN Example**
- **Schill, Springer, Hara**
- **Distributed Systems – SOA and Business Processes**
**Mapping BPMN to BPEL**

- **Bank**
  - Exclusive Gateway can be mapped to BPEL `switch`
  - Process Payment

- **Web Shop**
  - Parallel Gateway maps to BPEL flow
  - Purchasing
  - Verify Order
  - Analyse Risk
  - Calculate Discount
  - Handle Payment
  - Manage Invoicing
  - Arrange Shipping

- **Customer Management**
  - A task maps to a BPEL `invoke`
  - Manage Customer

- **Invoicing**
  - Gateway alternative maps to BPEL `case` in switch
  - Manage Invoicing

The entire set will be enveloped in a BPEL sequence.
Business Process Execution Language

- Language for Business Process Description as well as Composition of Web Services
- facilitates automated process integration
- standardised by OASIS as WS-BPEL (2007)
- based on XML and Web Services
- Syntax defined by XML schema
- no standardised graphical notation
Two kinds of business processes are definable

- **Abstract Process → Choreography**
  - specifies message exchanges between multiple participants
  - no definition of internal process details
  - Interfaces defined through set of all receive and reply

- **Executable Process → Orchestration**
  - specific activities and particular services which must be executed
  - executable via an execution engine

Composition of processes require a defined *conversation*. (i.e., context of the choreography or orchestration)
Orchestration vs. Choreography

**Orchestration**

1. Receive
2. Invoke
3. Invoke
4. … n: Invoke
5. Reply

**Choreography**

1. Invoke
2. Invoke
3. Reply
4. Invoke
5. Invoke
BPEL Process consists of activities

- **Primitive Activities** – for common tasks; i.e., invoke, receive, reply, wait, assign, throw, terminate, empty

- **Structured Activities** – to combine primitive activities; i.e., sequence, flow, switch, while, pick - select one of several alternative paths
Interaction between BPEL Processes through partnerLinks
- defined in WSDL files
- possess roles with portTypes e.g. invocation role: computePrice and callback role: provideInvoice

Description of conversations in correlationSets
- declarative description
- multiple concurrent conversations possible
BPEL–WSDL–XML Schema

```
<xs:schema>
  <xs:element/>
</xs:schema>
```

**BPEL**
```
<process>
  <partnerLinks/>
  <variables/>
  <scope>
    <assign/>
  </scope>
  ...
</process>
```

**WSDL**
```
<definitions>
  <partnerLinkTypes>
    <types/>
    <message>
      <part type../>
    </message>
    ...
  </definitions>
```
BPEL: Example Process

- **Purchasing**
  - portType: purchaseOrderPT
  - Operation: sendPO

- **BPEL-Process**
  - receive
  - reply

- **Shipping**
  - portType: shippingPT
  - Operation: requestShipping
  - invoke
  - receive

- **Invoicing**
  - portType: computePricePT
  - Operation: initiatePriceCalc
  - invoke
  - receive

- **Partner Link**
  - Partner Link

- **Partner Link**
  - Partner Link

- **Partner Link**
  - Partner Link

- **Partner Link**
  - Partner Link
BPEL: Example of a Process Step

BPEL File for Purchase Order Process

```xml
<process name="purchaseOrderProcess"/>
    <partnerLinks>
        ...
        <partnerLink name="invoicing" partnerLinkType="lns:invoicingLT"
                      myRole="invoiceRequester" partnerRole="invoiceService" />
    </partnerLinks>
    ...
    <sequence>
        ...
        <invoke partnerLink="invoicing"
                operation="initiatePriceCalc">
        ...
        <receive partnerLink="invoicing"
                 operation="sendInvoice">
        ...
    </sequence>
</process>
```

Source: [http://docs.oasis-open.org/wsbpel/2.0/CS01/wsbpel-v2.0-CS01.html](http://docs.oasis-open.org/wsbpel/2.0/CS01/wsbpel-v2.0-CS01.html)
BPEL-Engine: Runtime environment for the Interpretation of the BPEL documents und execution of the Processes

- IBM WebSphere Process Choreographer
- Microsoft BizTalk Server
- Oracle BPEL Process Manager
- Intalio n3 Server
- Apache Agila BPEL
- JBoss Application Server with jBPM
- and others...
WS-Extensions

Business Process

**WS-* Extensions**
(basic WS specification insufficient for comprehensive Business Processes)

Directory

Service description

Message format

Communication protocol

**WS-CDL / WSCI:** Web Services Choreography Description Language / Interface

**OWL:** Web Ontology Language

**RDF:** Resource Description Format

**SOAP**

**HTTP**

**TCP/IP**

**UDDI**

**WSDL**

**OWL-S, RDF-S**

**BPEL, WS-CDL, BPML, WSCI**

**Transaction**

**Coordination**

**Context**

**relatable comm.**

**Security**
Web Service Transaction support

- transactional behaviour also required for service-based business processes (cf. ACID criteria)

- Set of OASIS standards for Web Service Coordination
  - reach consistent agreement on the outcome of distributed activities (activities = computation units)
  - interoperability between execution environments
  - all participating services use a common transaction context

- Influence of transport protocols:
  - HTTP: single Request/Response-based message exchange
  - transaction context has to be mediated as part of each request
  - services have to follow protocol standards
  - asynchronous protocols require management of transaction context
Web Services Coordination (WS–C)

- extensible framework for coordinating activities and contexts
- Coordination in 3 Phases
  1. Activation (Creation of CoordinationContext)
  2. Registration (Web Services register as transaction participants)
  3. Coordination (of the results of the participating services)
1. Create Coordination Context

2. Context, Activity ID

3. addProductToDispatchList, CoordinatorContext

4. register, Activity ID

5. removeProductFromStorage, CoordinatorContext

6. register, Activity ID
Web Services Atomic Transactions (WS-AT)

- part of Web Service coordination framework
- coordination of short-lived distributed activities
- consists of three protocols for
  - completion: initiates commit processing
  - Volatile Two-Phase Commit for volatile resources
  - Durable Two-Phase Commit for durable resources
- specifies a coordinator, an initiator, and one or more participants

→ comprehensive processes require Web Service Business Activity (WS-BA) ⇒ atomic and mixed contexts, participant- and coordinator-based completion agreements
Example Business Activity

- Purchase action by customer initiates order
- Customer registered
- Prices calculated and payment booked
- Warehouse system updated but availability of one product is not according to customer requirements
- Compensation of already committed activities
  - Return products into warehouse system
  - Refund money to credit card
  - Change order or remove order from system
WS-Security

- OASIS Standard addressing major protection goals
  - Confidentiality (XML-Encryption)
  - Integrity (XML-Signature)
  - Accountability (XML-Signature)
  - Authenticity and Access Control (Security Assertions Markup Language (SAML))
- Foundation of further web service standards for security

```xml
... <soap:Header>
   <wsse:Security ...>
      <wsse:UsernameToken Id="me">
         <wsse:Username>Holtin</wsse:Username>
      </wsse:UsernameToken>
   </wsse:Security>
... </soap:Header> ...
```
WS-Security Framework

Extensions

Authentication
Username/Password, Kerberos, X.509

message integrity
XML Signature

message confidentiality:
XML Encryption

authorisation:
SAML-Profile

WS-Security Framework

Distributed Systems – SOA and Business Processes
Multiple security domains have different accounts for authentication

⇒ Single Sign On (SSO) required

**Approach**

- XML-based exchange language for authentication and authorisation
- Identity Provider (IdP) offers information about its subjects for other service providers (SP)
- IdP acts in the role of a Asserting Party (AP) and SP in the role of a Relying Party (RP)
- contract about the exchange of information necessary
SAML: Example for Single-Sign-On

- Message exchange according to SAML Web Browser SSO Profile
- HTTP POST binding mechanism and Authentication Request Protocol used to mediate SAML messages
Open Issues

- negotiation of mechanism between service user and service provider
- probability of renewal and validation of keys and identities for each message exchanged

WS-Security has its own extensions addressing the issues (extensible framework; WS-Security-*)
Technologies for Implementing a SOA

- **Enterprise Service Bus (ESB)**
  - based on message broker concepts
  - for loose coupling of mediated services
  - no standardisation

- **Java Business Integration (JBI)**
  - Java-based industry standard (JSR 208)
  - vendor-independent
  - supports existing standards (Java SE, Java EE, WSDL, WS-I Basic Profile, JMX)
  - considers scalability and performance in distributed deployment

- **Service Component Architecture**
  - extends ESB concepts while focusing on SOA concept
  - consists of a set of specifications for loose assembling on external interfaces
  - support of multitude of programming languages
Enterprise Service Bus: Example

Order Processing
- J2EE, .NET

Legacy Applications Customer Administration
- CORBA

Service Flow Specification (e.g. based on BPEL)

ESB – Message Broker
Common Runtime Environment

Warehouse Management

B2B Integration Payment Service

Shipping
JBI (Java Business Integration)-Concept:

- **Service Engines**: extensible business logic (e.g., EJB Wrapper)
- **Binding-Components**: Proxy for service users and remote services → access independent from transport protocol
- **Normalised Message Router**: exchange of (normalised) messages → interoperability between components
- **JBI runtime environment**
SCA: Specifications and Relations

External specifications

Generic specifications (technology-independent)

Technology or programming language specific specifications

Specifications released by Open Service Oriented Architectures (OSOA)
- Composition described in XML-based configuration file
- Composition is static, consists of concrete components, services, wires and property definitions
- Hierarchical composition possible
Service-oriented architectures focus on the creation of Business Processes
- Services encapsulate application logic
- Business processes are implemented with service compositions

Business Processes can involve many processing steps which involve various applications/services located on different servers

Language support for modelling business processes with mapping from business oriented description (BPMN) to executable process model (BPEL)

Service-oriented Business Processes require transactions and security
- Web Service Coordination
- Web Services Atomic Transactions (WS-AT)
- Web Service Business Activity (WS-BA)
- WS Security comprises standards for Confidentiality (XML-Encryption), Integrity (XML-Encryption), Accountability (XML-Signature) and Authenticity and Access Control (SAML)


BPMN 2.0 Engine – Activiti: http://activiti.org/userguide/index.html

OASIS Standard: Web Services Coordination (WS-Coordination), 2007

OASIS Standard: Web Services Atomic Transaction, 2007

OASIS Standard: Web Services Business Activity

(WS-BusinessActivity) Version 1.1, 16 April 2007

Ten-Hove, R., Walker, P.: Java Business Integration 1.0, JSR-208,

OSOA Home Page: www.osoa.org