Distributed Systems

Seminar 9 – SOA and Business Processes

Dr. Ing. Thomas Springer
Technische Universität Dresden
Chair of Computer Networks
Business Process Modeling

The BPMN diagram below models a simple business process of an insurance company to handle a contract request and prepare an insurance contract.

a) Extend the BPMN diagram with an additional activity “Register Customer” performed in parallel to the “Preverification”, “Special Audit” and “Calculate Surcharge” activities by the “Customer Management” department of the Insurance Company.

b) Sketch the extended business process using Petri Nets.

c) Extend the Petri Net with an additional activity “Send Contract Online” performed as alternative to “Send Contract” by the “Contract Handling Department”.

d) Add an activity “Archive Contract” which should be performed in parallel to “Send Contract” and “Send Contract Online” by the “Contract Handling Department”.

Exercise 9.1
Solution E9.1a

[Diagram with process flow for contract request handling and auditing, including steps like register contract request, pre-verification, special audit, calculate surcharge, issue contract, and send contract.]

Dr. Thomas Springer
Solution E9.1b

special audit

calculate surcharge

issue contract

send contract
Solution E9.1c

special audit

calculate surcharge

send contract online

send contract
Solution E9.1d

- special audit
- calculate surcharge
- send contract online
- send contract
- archive contract
Service-oriented Business Processes

a) Propose a set of services which can be used to map the business process discussed in 9.1 to an executable process.

b) Sketch the BPEL process corresponding to the extended business process discussed in 9.1. Name the relevant structured activities for modeling the structure of the business process.

c) Give an example for the definition of the mapping of a process activity to a service based on the concept of partner links.
- ContractHandlingService
  - registerContractRequest(contractRequest)
- AuditingService
  - Preverify(contractRequest)
  - CalculateSurcharge(contractRequest)
  - PerformSpecialAudit(contractRequest)
- CustomerManagementService
  - registerCustomer(customerData)
- ContractIssuingService
  - Contract issueContract(contractRequest)
- DispatchService
  - sendDocument(document)
  - sendDocumentOnline(document)
- ArchivingService
  - archiveDocument(document)
<process>
  <partnerLinks /> ...
  <variable name="contractRequest" type=.../>
</sequence>
  <receive />
  <invoke partnerLink="ContractHandling" operation="registerContractRequest" inputVariable="contractRequest" />
</flow>
  <sequence>
    <invoke partnerLink="Auditing" operation="preverify"/>
    <switch>
      <case condition="special" >
        <invoke partnerLink="Auditing" operation="performSpecialAudit"/>
      </case>
      <case condition="surcharge" >
        <invoke partnerLink="Auditing" operation="calculateSurcharge"/>
      </case>
      <else/>
    </switch>
  </sequence>
  <invoke partnerLink="CustomerManagement" operation="registerCustomer"/>
</flow>
...
...<invoke partnerLink="ContractIssuing" operation="issueContract" />
<flow>
  <switch>
    <case condition="online">
      <invoke partnerLink="Dispatch" operation="sendDocumentOnline" />
    </case>
    <case condition="offline">
      <invoke partnerLink="Dispatch" operation="sendDocument" />
    </case>
    <else/>
  </switch>

  <invoke partnerLink="Archiving" operation="archiveDocument"/>
</flow>
</sequence>
</process>
A partner link basically maps to a WSDL web service 'portType', so one partnerLink (e.g. 'printService' above) maps to a single web service.

**BPEL**

```xml
<process>
    <import importType="http://schemas.xmlsoap.org/wsdll/"
        location="../../test_bucket/service_libraries/tptp_EnginePrinterPort.wsdl"
        namespace="http://www.eclipse.org/tptp/choreography/2004/engine/Print" />

    <partnerLinks>
        <partnerLink name="ContractHandling" partnerLinkType="ContractHandlingLT"
            myRole="RegisterRequester"
            partnerRole="RegistrationService"/>
    </partnerLinks>

    ...

    <invoke partnerLink="ContractHandling"
        operation="registerContractRequest" inputVariable="contractRequest" />

    ...

</process>
```
WSDL for ContractHandlingService

<message name="ContractRequest">
    <part name="value" type="xsd:string"/>
</message>

<portType name="ContractHandlingPort">
    <operation name="registerContractRequest">
        <input message="tns:ContractRequest"/>
    </operation>
</portType>

<partnerLinkType name="ContractHandlingLT">
    <role name="RegistrationService" portType="tns:ContractHandlingPort"/>
</partnerLinkType>
Web-Service Standards

a) What is specified with the following standards: WS-Coordination, WS-AtomicTransaction, WS-BusinessActivity, XML Encryption, XML Signature, WS-Security, SAML

b) What is the benefit of using XML Encryption to encrypt parts of SOAP messages?
WS-Coordination

- Extensible framework for coordinating activities
  - Central, generic WS for transaction coordination
    - Activation service
    - Registration service
  - Several protocols can be plugged in

WS-AtomicTransaction

- Adoption of known concepts to Services
- Transaction management and commit based on 2PC protocol

WS-BusinessActivity

- Protocols for managing long-term transactions called business activities
- Includes compensation and mixed outcome
XML Encryption
- defines how to encrypt the contents of an XML element
- XML element refers to the cipher text, keying information, and algorithms

XML Signature
- XML syntax for digital signatures
- Signature and meta-information (method, key info, ...) is represented in XML

WS-Security
- enhancements to SOAP messaging
  - message integrity
  - message confidentiality
  - Accountability – signatures
  - Authenticity and Access Control - single message authentication
- Based on other standards like XML Signature, XML Encryption and SAML
SAML - Security Assertion Markup Language

- XML-based standard for **exchanging authentication and authorization data between security domains**, that is, between an *identity provider* (a producer of assertions) and a *service provider* (a consumer of assertions).
XML Encryption

- Flexible routing based on SOAP-Header information requires access to SOAP header
- End-to-end encryption for message content
- While SOAP body is encrypted

- Parts of SOAP-body for content-based routing
Technologies for Implementing a SOA

a) Name the key features of an Enterprise Service Bus and the Service Component Architecture.

b) What problem domains address both approaches?
Enterprise Service Bus

Domain
- Integration of heterogeneous IT-Infrastructures
- Loose coupling bases on Message Routing

Main Features
- Key concept: message broker – the bus
  - Communication via message-oriented middleware based on standard set of messages
  - Abstraction for endpoints (technology independence)
  - Support of many transport media and standard protocols
  - Routing and transformation of messages

additional features provided:
- security services, single-sign-on
- service registry
- data conversion and mapping services
- reliable message transport (based on MOM)
- central monitoring
- scalability, performance, transaction support
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

- SOAP
  - Warehouse Management

- JMS
  - B2B Integration
  - Payment Service

- SOAP
  - Shipping

Dr. Thomas Springer
Service Component Architecture

Domain:
- Integration of different services and systems
- Reducing the development effort for SOA infrastructures

Main Features
- Loose coupling of technology independent services (Assembly model)
- Mapping of abstract assemblies to different Programming languages/platforms
- Composition is static, consists of concrete component, service, wire and property definitions
- Hierarchical composition possible
- Composition described in XML-based configuration file
- wires connect services
- composition is static, consists of concrete component, service, wire and property definitions
- hierarchical composition possible